

DAS COMPUTERUNTERSTÜTZTE ENTSCHEIDUNGSTRAINING

- KONZEPT, EXPERIMENTE UND ANFORDERUNGSPROFIL -

von

Mag. rer. soc. oec. H. Ö. Hubert Österle

aus

Dornbirn

1973



Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41548408_0088

DAS COMPUTERUNTERSTÜTZTE ENTSCHEIDUNGSTRAINING

- KONZEPT, EXPERIMENTE UND ANFORDERUNGSPROFIL -

Inaugural-Dissertation

zur Erlangung des akademischen Grades

eines Doktors der Wirtschafts- und Sozialwissenschaften

(Dr. rer. pol.)

an der Wirtschafts- und Sozialwissenschaftlichen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

vorgelegt

von

Mag. rer. soc. oec. H. Ö. Hubert Österle

AUS

Dornbirn



Rerferent: Prof. Dr. P. Mertens

Korreferent: Prof. Dr. R. Berke

Tag der mündlichen Prüfung: 7. November 1973

G l i e d e r u n g

=====

Übersicht über den Aufbau der Arbeit	1
Teil 1: Analyse der Ausbildungsmethoden der betriebswirtschaftlichen Entscheidungslehre	2
1. Lernziele der betriebswirtschaftlichen Entscheidungslehre	2
1.1. Stoffgebiet	2
1.2. Lernzielniveau	4
2. Erfüllung der Lernziele durch die vorhandenen Methoden	7
2.1. Lernzielstufe 1: Modell beschreiben	7
2.1.1. Literaturstudium	7
2.1.2. Programmierter Unterricht	7
2.1.3. Rechnerunterstützter Programmierter Unterricht (RPU)	8
2.1.4. Rechnerunterstützte Informationssuche	8
2.1.5. Rechnerunterstützte Selbsttests	10
2.1.6. Vorlesung	10
2.2. Lernzielstufe 2: Modell anwenden	11
2.2.1. Übung	11
2.2.2. Rechnerunterstützte Übung	12
2.2.3. Fallstudie	12
2.2.4. Unternehmensplanspiel	12
2.2.5. Problemlösung mit interaktiven Programmen	14
2.3. Lernzielstufe 3: Modell modifizieren	16
3. Zusammenfassung und Schlußfolgerung: Lücken im vorhandenen Angebot von Lehrmethoden	17

Teil 2: Konzeption der Methode CET

18

1.	Einführendes Beispiel: Darstellung einer Terminalsitzung	18
2.	Der Begriff "Computerunterstütztes Entscheidungstraining"	41
3.	Charakteristika des CET	41
3.1.	Simulation der Praxis	41
3.2.	Nutzung des Computers	42
3.3.	Lehrmethoden-"Mix"	43
3.4.	Freiheit des Lernenden	44
3.5.	Lerndatenerhebung	46
4.	Lehrstrategien des Computerunterstützten Entscheidungstrainings	46
4.1.	"Off-line-Studium"	48
4.1.1.	Literaturstudium	48
4.1.2.	Lehrveranstaltungen	48
4.2.	Informationssuche	49
4.2.1.	Aufgaben	49
4.2.2.	Realisierung	49
4.3.	Rechnerunterstützter Programmierter Unterricht	50
4.3.1.	Aufgaben	50
4.3.2.	Realisierung	50
4.4.	Rechnerunterstützte Selbsttests	54
4.4.1.	Aufgaben	54
4.4.2.	Realisierung	55
4.5.	Training	56
4.5.1.	Aufgaben	56
4.5.2.	Realisierung	56
4.6.	Problemlösung	57
4.6.1.	Aufgaben	57
4.6.2.	Realisierung	57
4.7.	Lernersteuerung	57
4.7.1.	Aufgaben	57
4.7.2.	Realisierung	58

5.	Erfahrungen mit dem CET	59
5.1.	Lernerfolg	60
5.2.	Beurteilung des CET durch den Studenten	61
5.3.	Analyse des CET	62
5.4.	Zusammenfassung	66

Teil 3: Realisierungsexperimente	67
----------------------------------	----

1.	Softwaresysteme zur Erstellung von CET- Programmen	67
1.1.	Unterrichtssysteme	68
1.2.	On-line-Programmiersysteme	68
1.3.	Off-line-Programmiersysteme	69
2.	Experimente	69
2.1.	PLANIT	69
2.1.1.	Didaktogramm	70
2.1.2.	Programm	73
2.1.3.	Dialogbeispiel	80
2.1.4.	Zusammenfassung	92
2.2.	PUMA	93
2.2.1.	Didaktogramm	93
2.2.2.	Programm	97
2.2.3.	Dialogbeispiel	102
2.2.4.	Zusammenfassung	106
2.3.	APL	107
2.3.1.	Didaktogramm	107
2.3.2.	Programm	108
2.3.3.	Dialogbeispiel	109
2.3.4.	Zusammenfassung	115
2.4.	TS	115
2.4.1.	Didaktogramm	115
2.4.2.	Programm	121
2.4.3.	Dialogablauf	127
2.4.4.	Zusammenfassung	146

Teil 4: Anforderungsprofil und Realisierungsmöglichkeiten	147
1.	Anforderungen an die Hardware 148
1.1.	Studentenplatz 149
1.1.1.	Dialog-Ein-/Ausgabeeinheit 149
1.1.1.1.	Ausgabe 149
1.1.1.2.	Eingabe 152
1.1.2.	Umgebung des Terminals 153
1.2.	Rechner 153
1.2.1.	Internspeicherkapazität 153
1.2.2.	Rechengeschwindigkeit 154
1.3.	Externspeicher 155
1.4.	Ein-/Ausgabe-Peripherie (außer Dialogstationen) 156
1.5.	Zusammenfassung 156
2.	Anforderungen an die Software 156
2.1.	Grundsätzliche Entscheidungen 157
2.1.1.	Mehrere Teilsysteme oder allgemeines MMK-System? 157
2.1.2.	Compilierendes oder interpretierendes System? 158
2.1.3.	Maschinenabhängiges oder -unabhängiges System? 159
2.1.4.	Isolierter Adressatenplatz oder Möglichkeit zum Zusammenschluß von Adressatenplätzen? 159
2.2.	Systemaufbau 160
2.2.1.	Baukastenprinzip 160
2.2.2.	Integration der Strategien 160
2.2.3.	Routinedefinitionen 161
2.2.4.	Trennung von Text, Struktur und Rechenroutinen 161
2.3.	Systemteile 162
2.3.1.	Struktur 163
2.3.1.1.	Aufgaben 163

2.3.1.2.	Realisierung in vorhandenen Systemen	164
2.3.1.2.1.	PLANIT	164
2.3.1.2.2.	PUMA	166
2.3.1.2.3.	TS	167
2.3.1.2.4.	Geplanter Ausbau des TS	167
2.3.2.	Rechenroutinen	167
2.3.2.1.	Aufgaben	167
2.3.2.2.	Realisierung in vorhandenen Systemen	169
2.3.2.2.1.	PLANIT	169
2.3.2.2.2.	PUMA	170
2.3.2.2.3.	TS	171
2.3.2.2.4.	Geplanter Ausbau des TS	172
2.3.3.	Ausgabeaufbereitung	172
2.3.3.1.	Aufgaben	172
2.3.3.1.1.	Bestimmung des Ausgabemediums	172
2.3.3.1.2.	Ausgabearten	173
2.3.3.1.2.1.	Textkonserven	173
2.3.3.1.2.2.	Variable Inhalte	174
2.3.3.1.2.2.1.	Skalare	174
2.3.3.1.2.2.2.	Vektoren	174
2.3.3.1.2.2.3.	Darstellung von Graphiken	178
2.3.3.2.	Realisierung in vorhandenen Systemen	179
2.3.3.2.1.	PLANIT	179
2.3.3.2.2.	PUMA	189
2.3.3.2.3.	TS	181
2.3.3.2.4.	Geplanter Ausbau des TS	182
2.3.4.	Antwortverarbeitung	182
2.3.4.1.	Aufgaben	182
2.3.4.1.1.	Numerische Ausdrücke	183
2.3.4.1.2.	Algebraische Ausdrücke	183
2.3.4.1.3.	Auswahlantworten	183
2.3.4.1.4.	Verbale Ausdrücke	183
2.3.4.1.5.	Befehle	184
2.3.4.2.	Realisierung in vorhandenen Systemen	185

2.3.4.2.1.	PLANIT	185
2.3.4.2.2.	PUMA	186
2.3.4.2.3.	TS	187
2.3.4.2.4.	Geplanter Ausbau des TS	188
2.3.5.	Befehlsinterpreter	188
2.3.5.1.	Aufgaben	188
2.3.5.1.1.	Globalsteuerung	189
2.3.5.1.2.	Dialogablaufsteuerung	190
2.3.5.1.3.	Aufruf von Systemdienstleistungen	191
2.3.5.1.4.	Modifikation von Parametern	192
2.3.5.2.	Realisierung in vorhandenen Systemen	192
2.3.5.2.1.	PLANIT	193
2.3.5.2.2.	PUMA	194
2.3.5.2.3.	TS	195
2.3.5.2.4.	Geplanter Ausbau des TS	195
2.3.6.	Datenbereitstellung	195
2.3.6.1.	Aufgaben	195
2.3.6.2.	Realisierung in vorhandenen Systemen	197
2.3.6.2.1.	PLANIT	198
2.3.6.2.2.	PUMA	198
2.3.6.2.3.	TS	198
2.3.6.2.4.	Geplanter Ausbau des TS	199
2.3.7.	Lerndatenaufzeichnung und -auswertung	199
2.3.7.1.	Aufgaben	199
2.3.7.2.	Realisierung in vorhandenen Systemen	203
2.3.7.2.1.	PLANIT	203
2.3.7.2.2.	PUMA	204
2.3.7.2.3.	TS	204
2.3.7.2.4.	Geplanter Ausbau des TS	204
2.3.8.	Eingabe des Kursmaterials	205
2.3.8.1.	Aufgaben	205
2.3.8.2.	Realisierung in vorhandenen Systemen	206
2.3.8.2.1.	PLANIT	207
2.3.8.2.2.	PUMA	207
2.3.8.2.3.	TS	207
2.3.8.2.4.	Geplanter Ausbau des TS	207
2.4.	Zusammenfassung	208

Fünfter Teil:	Ausblick auf den Einsatz des CET in der Praxis	215
1.	Einsatz in der betriebswirtschaft- lichen Ausbildung an der Universität	215
1.1.	Einsatz des CET für die Ausbildung im Fach Betriebsinformatik	215
1.1.1.	Lehrveranstaltung "Simulation"	216
1.1.2.	Lehrstoff	222
1.2.	Schwierigkeiten beim Einsatz des CET an der Universität	223
2.	Einsatz in der berufsbegleitenden Weiterbildung	224
Literaturverzeichnis		226

Übersicht über den Aufbau der Arbeit

Ziel dieser Arbeit ist es, ein Konzept für die Lehrmethode "Computerunterstütztes Entscheidungstraining" (CET) zu entwerfen und Möglichkeiten für eine Realisierung zu untersuchen. Dies soll zu einem Anforderungsprofil an die Hard- und Software für die Verwirklichung des CET führen.

Im ersten Teil wird die derzeitige Ausbildungssituation der betriebswirtschaftlichen Entscheidungslehre kritisch beleuchtet und daraus das Einsatzgebiet des CET abgeleitet.

Im zweiten Teil wird diese Lehrmethode an einem Beispiel demonstriert. Danach werden ihre Charakteristika und der Aufbau des Methoden-"Mixes" erläutert. Die Beschreibung wird anhand einiger Ergebnisse aus einer Studentenforschung ergänzt.

Der dritte Teil zeigt einige der Experimente mit verschiedenen Softwaresystemen, die einerseits zur Auswahl des geeigneten Systems nötig waren und andererseits gemeinsam mit dem Konzept des CET und den Ergebnissen der Studententests die Grundlage für das Anforderungsprofil darstellen.

Dieses wird im vierten Teil gemeinsam mit den Realisierungsmöglichkeiten in ausgewählten Systemen beschrieben. Der Katalog der Anforderungen soll helfen, bei zukünftigen Softwareentwicklungen Anwendungen wie das CET zu berücksichtigen.

Die Arbeit schließt im fünften Teil mit einem kurzen Ausblick auf mögliche Einsatzgebiete des CET.

E r s t e r T e i l

Analyse der Ausbildungsmethoden der

betriebswirtschaftlichen Entscheidungslehre

1. Lernziele der betriebswirtschaftlichen Entscheidungslehre

1.1. Stoffgebiet

Die betriebswirtschaftliche Entscheidungslehre soll dem Studenten die Kenntnis jener Methoden vermitteln, die er nach Abschluß des Universitätsstudiums bei der Entscheidungsvorbereitung im Betrieb benötigt. Zur Veranschaulichung der Art des Lernstoffes werden einige Teilgebiete durch Beispiele erläutert:

- Allgemeine Entscheidungshilfsmittel

Als Beispiel sei hier auf das Entscheidungsbaumverfahren hingewiesen, das mit Hilfe der Roll-back-Analyse die Untersuchung mehrstufiger Entscheidungssituationen unterstützt¹⁾.

- Prognoserechnung

Die betriebswirtschaftliche Prognoserechnung beschäftigt sich hauptsächlich mit der kurz- und mittelfristigen Vorhersage. Dazu werden Verfahren wie beispielsweise die schrittweise Regression mit adaptiver Einflußgrößenkombination²⁾ verwendet. Diese beruht auf dem Prinzip der Regres-

1) Vgl. Magee, J.F., Decision Trees for Decision Making, Harvard Business Review 42 (1964) 4, S. 126 ff.

2) Vgl. dazu Griese, J. und Matt, G., Prognose mit Hilfe einer Kombination von schrittweiser Regressionsanalyse und exponentieller Glättung (adaptive Einflußgrößenkombination), in: Mertens, P. (Hrsg.), Prognoserechnung, Würzburg-Wien 1973, S. 159 ff.

sionsrechnung und benutzt die Teilverfahren Exponentielles Glätten, Lineare Planungsrechnung und die Methode der kleinsten Quadrate sowie den t-Test zur Erstellung und Überprüfung des Regressionsansatzes.

- Investitionsplanung

Die Ermittlung von günstigen Investitionsbudgets kann durch formale Verfahren wie beispielsweise die lineare Planungsrechnung in Verbindung mit Sensitivitätsanalysen und durch verschiedene Bewertungsverfahren unterstützt werden¹⁾. Mit Hilfe einer Kombination von Risiko- und Break-even-Analyse können einzelne Investitionsalternativen auf ihre Rentabilität bei gegebenem Produktionsniveau untersucht werden²⁾.

- Simulation

Die Simulation versucht durch Abbildung der Wirklichkeit in Modellen das Verhalten von Systemen zu erklären und gegebenenfalls Optima zu finden. Sie verwendet dazu eine Reihe von Hilfsverfahren wie zum Beispiel die Methode des steilsten Anstieges, die der Einschränkung des Simulationsumfanges dient³⁾.

- Datenverarbeitung

Aus dem breiten Stoffgebiet der Datenverarbeitung sei nur die Mensch-Maschine-Kommunikation erwähnt, bei der Mensch und Maschine im Dialog stehen und jeder für ihn spezifische Tätigkeiten verrichtet⁴⁾.

1) Vgl. Mertens, P., Griesse, J., Industrielle Datenverarbeitung, Bd. 2, Informations- und Planungssysteme, Wiesbaden 1972, S. 183 ff.

2) Vgl. Rackelmann, G., Einsatz der Simulation bei der Break-even-Analyse, Diplomarbeit Nürnberg 1973.

3) Vgl. Mertens, P., Simulation, Stuttgart 1968, S. 31 ff.

4) Vgl. Mertens, P. und Kress, H., Mensch-Maschinen-Kommunikation als Hilfe bei der Entscheidungsvorbereitung und Planung, Zeitschrift für betriebswirtschaftliche Forschung 22 (1970) 1, S. 1 f.

Allgemein kann man sagen, daß die betriebswirtschaftliche Entscheidungslehre dem Studenten zeigen soll,

- welche Entscheidungsprobleme in der Unternehmung auftauchen,
- welche methodischen Hilfsmittel und Modelle für die Entscheidungsvorbereitung vorhanden sind,
- wie diese aufgebaut sind,
- wie sie zur Entscheidungsfindung herangezogen werden können und
- wie der Computer als Hilfe bei der Problemlösung eingesetzt werden kann.

1.2. Lernzielniveau

Das Lernzielniveau, das in der betriebswirtschaftlichen Ausbildung angestrebt werden muß, kann am besten anhand eines einfachen Beispiels erläutert werden. Wir verwenden dazu die exponentielle Glättung erster Ordnung, ein mittelwertbildendes Verfahren, das eine unterschiedliche Gewichtung der Daten erlaubt und somit besonders für Prognosezwecke geeignet ist.

Das Globalziel für die Lehre dieses Verfahrens könnte lauten: Der Student soll die exponentielle Glättung so beherrschen, daß er sie sinnvoll einsetzen kann.

Dafür ist die Erreichung folgender Teillernziele notwendig:

- (1) Der Student muß wissen, wie groß der Glättungsparameter gewöhnlich ist.
- (2) Er muß mit dem Verfahren rechnen können.
- (3) Er muß die Auswirkungen der Größe des Glättungsparameters kennen.

- (4) Er muß einen sinnvollen Ausgangswert für die geglättete Größe bestimmen können.
- (5) Er muß zu einer beliebigen Datenkonstellation einen Glättungsparameter angeben können, der die Besonderheiten der jeweiligen Daten berücksichtigt.
- (6) Er muß das Verhalten des Modells bei
 - einem positiven oder negativen Trend,
 - einem saisonalen Verlauf und
 - einem Trendbruchkennen.

Versucht man, die angegebenen Lernziele in die Bloom'sche Taxonomie¹⁾ (Wissen, Verstehen, Anwenden, Analyse, Synthese und Bewertung) einzuordnen, so kann man Lernziele (1) der Kategorie "Wissen", Lernziele (2) und (3) der Lernzielkategorie "Verstehen", (4) und (5) der Kategorie "Anwenden" und (6) der Kategorie "Analysieren" zuordnen.

Sollen außer Betriebswirten, die vorhandene Verfahren in der Praxis anwenden müssen, auch Operations-Research-Spezialisten unterrichtet werden, so muß man noch höhere Lernziele fordern. Diese Studenten sollen in die Lage versetzt werden, selbständig Verfahren zu ändern.

Wenn diese Lernziele im Zusammenhang mit der Ausbildung in der Betriebswirtschaftslehre betrachtet werden, kann man sie zu folgenden drei Gruppen zusammenfassen:

- Stufe 1: Modell beschreiben
- Stufe 2: Modell anwenden
- Stufe 3: Modell modifizieren

Ohne Zweifel muß der Betriebswirtschaftsstudent primär mit der Beschreibung und Anwendung von Modellen in der Praxis vertraut gemacht werden, während die Modifikation der Modelle

1) Achtenhagen, F., Taxonomie von Lernzielen, in: Willmann-Institut (Hrsg.), Lexikon der Pädagogik, Bd. 4, Freiburg 1971, S. 209 f.

von einer zahlenmäßig eher kleinen Gruppe der Studenten erreicht werden soll.

Dies kann man auch aus der Aussage von Hahn entnehmen, in der er von den Führungskräften fordert, daß sie "in der Lage sind,

- Probleme in wirtschaftlichen, technischen und/oder sozialen Dimensionen im Hinblick auf die jeweils relevante Zielsetzung klar zu erkennen.
- Lösungsmöglichkeiten für die Probleme aufzuzeigen und sie zu bewerten sowie
- begründet eine im Hinblick auf das Ziel bestmögliche Lösung zu finden und durchzusetzen" ¹⁾.

Er folgert daraus:

"Die Ausbildung von Führungskräften erfordert hiernach vor allem

- (1) Wissensvermittlung und
- (2) Wissensanwendungstraining" ¹⁾.

Der Student muß nach Abschluß seines Universitätsstudiums wissen, welche Verfahren zur Unterstützung der Entscheidungsfindung in Betracht kommen. Er soll sie aber auch selbst einsetzen oder zumindest deren Ergebnisse interpretieren können. Wird ihm dieses Wissen auf der Universität nicht vermittelt, so kann er einerseits nicht beurteilen, welche Verfahren für eine konkrete Entscheidungssituation im Unternehmen geeignet sind, und kann sie andererseits nicht gezielt einsetzen. Er ist dann gezwungen, seine Anwendungserfahrung im Unternehmen zu sammeln. Dies hat zwar den Vorteil, daß er sich nur mit jenen Modellen beschäftigen muß, die er tatsächlich braucht, wirkt sich aber für das Unternehmen nachteilig aus, da sich Fehlentscheidungen in der Praxis unmittelbar in den Kosten niederschlagen, während sie beim Training am Modell keine finanziellen Auswirkungen haben.

1) Hahn, D., Allgemeine Bedeutung der Fallmethode für die Ausbildung an Hochschulen und in Seminaren für Führungskräfte, in: Alewell, K., Bleicher, K., Hahn, D., Entscheidungsfälle aus der Unternehmenspraxis, Wiesbaden 1971, S. 15.

2. Erfüllung der Lernziele durch die vorhandenen Methoden

In diesem Punkt soll die Eignung der vorhandenen Ausbildungsmethoden für die in Punkt 1 aufgestellten Lernziele untersucht werden. Es wird dabei den neueren Versuchen zur Verbesserung der Ausbildung besonderes Augenmerk geschenkt.

Die Eignung der Ausbildungsverfahren ist außer von lernerbezogenen Faktoren auch vom Lehrstoff und dem Lernzielniveau abhängig. Da wir den Lehrinhalt durch die Eingrenzung auf die Entscheidungslehre bereits auf ein relativ homogenes Stoffgebiet eingeschränkt haben, ist vor allem die Eignung für die verschiedenen Lernzielstufen zu untersuchen. Wir fassen dabei für jede Lernzielebene jene Verfahren zusammen, die u. E. am ehesten dafür geeignet sind und auch in der Unterrichtspraxis entsprechend eingesetzt werden.

2.1. Lernzielstufe 1: Modell beschreiben

2.1.1. Literaturstudium

Skripten, Lehrbücher, Zeitschriften und andere Fachliteratur sind kostengünstige Medien zur Vermittlung von Grundwissen. Diese Lehrmittel haben zwar für den Studierenden oft den großen Nachteil, didaktisch äußerst mangelhaft zu sein, bieten ihm aber die Möglichkeit, Lernwege und Lerngeschwindigkeit selbst zu bestimmen.

2.1.2. Programmierter Unterricht

Als Sonderfall des Literaturstudiums könnte man den Programmierten Unterricht bezeichnen. Er dürfte vor allem als Einführung in komplexe Stoffgebiete geeignet sein und wird bereits häufig dafür eingesetzt¹⁾.

1) Vgl. dazu Mertens, P. und Plötzeneder, H.D., Ein Experiment mit Programmierter Unterricht im betriebswirtschaftlichen Hochschulstudium, Zeitschrift für Betriebswirtschaft 41 (1971) 3, S. 188 und 208.

2.1.3. Rechnerunterstützter Programmierter Unterricht (RPU)¹⁾

Den größten Anteil am Rechnerunterstützten Unterricht (RUU)¹⁾ hat derzeit der Programmierter Unterricht (PU)¹⁾. Er wird jedoch für die betriebswirtschaftliche Ausbildung bisher sehr selten eingesetzt. Dies mag größtenteils auf die relativ hohen Kosten für die Erstellung und Darbietung der Kurse sowie darauf zurückzuführen sein, daß der RPU keine deutlich besseren Ergebnisse als der PU in Buchform liefert.

Bei Versuchen mit RPU hat sich gezeigt, daß die Lernenden den Einsatz von derartigen Kursen sehr unterschiedlich beurteilen. Es bestehen zwei Gruppen, von denen die eine den RPU klar ablehnt und die andere ebenso klar befürwortet²⁾. Rockart hat bei einer Studentenbefragung festgestellt, daß der Großteil der Studenten den Einsatz eines bestimmten RPU-Programmes ablehnt, eine zahlenmäßig eher kleine Minderheit diese Lehrmethode aber vorzieht³⁾. Dieselbe Tendenz ist auch bei Befragungen des Projektes CET⁴⁾ zum Ausdruck gekommen.

Rockart stellt zum RPU fest:

"The automation of traditional programmed instruction texts is, at best, unexciting for the learning process and of dubious cost-effectiveness value in relation to paper versions of the same material"⁵⁾.

2.1.4. Rechnerunterstützte Informationssuche

Zur Vermittlung von Faktenwissen, insbesondere zu Ergänzungszwecken, kann auch ein rechnerunterstütztes Informationserschließen

1) Da wir noch weiter diesen etwas langen Ausdruck benutzen müssen, führen wir dafür die angegebene Abkürzung ein.

2) Vgl. Dixon, G.T., Analyzing Student Performance to Improve an Interactive Teaching Program, Massachusetts Institute of Technology (MIT), MS-Thesis Cambridge 1970, S. 29 ff.

3) Vgl. Rockart, J.F., o.T., MIT, Arbeitspapier Cambridge 1972, S. 29 ff.

4) Vgl. Teil 2, Punkt 5.

5) Rockart, J.F., a.a.O., S. 29.

Bungs-(Information Retrieval-)system verwendet werden¹⁾.

Ein System zur Beantwortung von Schülerfragen wurde im Rahmen des "Associative Learning Project" am MIT unter dem Namen ALPS erstellt²⁾. Der Student kann mit Hilfe verschiedener Operatoren gezielt Informationen aufsuchen. Er gibt seinen Suchbegriff und einen der folgenden Suchoperatoren gemeinsam ein:

- DEF: Der Student will eine Definition eines Begriffes.
- USE: Bei Eingabe von USE wird die Verwendung des Begriffes erklärt.
- WHEN: Der zeitliche Aspekt des Begriffes wird näher erläutert.
- WHERE: Durch den Operator WHERE wird ausgegeben, wo dieser Begriff benutzt wird.
- EXPL: Einige Beispiele zur Erläuterung des Begriffes werden ausgegeben.
- REL: Die Beziehung zu anderen Begriffen wird erklärt.

Für all diese Anfragen muß natürlich eine Antwort vom Autor vorgesehen werden. Es sind in einem derartigen Informationssystem je nach Zielsetzung des Systems neben den hier erwähnten Suchoperatoren auch andere denkbar.

Das System ALPS wurde dem Studenten als Ergänzung bei der Durcharbeitung eines RPU-Kurses angeboten. Es hat sich gezeigt, daß die Anzahl der Zugriffe auf ALPS mit der Erhöhung der Qualität des PU abgenommen hat.

Die Problematik derartiger Systeme liegt neben den hohen Verarbeitungs- und Speicherkosten vor allem darin, daß derzeit die Studenteneingabe nur syntaktisch analysiert werden kann, eine semantische Analyse aber noch vollkommen fehlt. Zinn, der derartige Systeme in die Gruppe "Provision for conversation

1) Vgl. dazu auch Haefner, K., Status und Zukunft des Computer-unterstützten Hochschulunterrichts in Naturwissenschaften und Medizin, Arbeitspapier Freiburg 1972, S. 21 ff.

2) Vgl. Rockart, J.F., Morton, M.S.S., Zannetos, Z.S., Associative Learning Project - Phase 1 System, MIT, Working Paper 440-70 Cambridge 1970.

within a limited context" einreicht, lehnt die Erstellung von Programmen, die eine verbale Konversation von Schüler und Computer erlauben, ab, solange nicht ein Fortschritt bei der Verarbeitung natürlicher Sprachen erzielt wird, der es wenigstens erlaubt, die Eingabe des Studenten zu klassifizieren¹⁾".

2.1.5. Rechnerunterstützte Selbsttests

Dem Studenten wird die Möglichkeit geboten, sich einem Wissenstest auf einem bestimmten Gebiet zu unterziehen²⁾. Diese Selbsttests übernehmen zwar keine direkten Lehrfunktionen, können aber wertvolle Hinweise auf bestehende Wissenslücken geben und somit indirekt die Lernwirksamkeit erhöhen.

Die Kosten für die Erstellung und Durchführung von rechnerunterstützten Selbsttests liegen wesentlich niedriger als bei anderen rechnerunterstützten Lehrstrategien, da sie einerseits nicht sehr umfangreich sein müssen und andererseits die spezifischen Fähigkeiten des Computers besser nutzen.

2.1.6. Vorlesung

"Die Vorlesung ist mit durchschnittlich 46 % Anteil an der gesamten Lehrzeit nach wie vor die am stärksten verwendete Lehrveranstaltungsform³⁾. Dies ist eines der Ergebnisse einer Untersuchung über betriebswirtschaftliche Hochschuldidaktik im deutschsprachigen Raum. Allerdings wird in demselben Bericht angemerkt, daß unter diesem Begriff nicht mehr die streng monologartige Vorlesung zu verstehen ist. Dies entspricht auch den gängigen Interpretationen des Inhalts dieser

1) Vgl. Zinn, K., Requirements for Programming Languages in Computer-Based Instructional Systems; Technical Report; Organization for Economic Co-operation and Development (OECD); Centre for Educational Research and Innovation, CERI/CT/70.62, Paris 1971, S. 50.

2) Vgl. Haefner, K., a.a.O., S. 16.

3) Mertens, P., Ergebnisse einer Umfrage zur Verbreitung einzelner Lehr- und Prüfungsformen, in: Mertens, P. (Hrsg.), Betriebswirtschaftliche Hochschuldidaktik, Wiesbaden 1972, S. 23.

Lehrveranstaltungsform, in denen von der Vorlesung nicht die Vermittlung von Grundlagenwissen, das auch durch Literaturstudium erworben werden kann, sondern die aktuelle Ergänzung und die Anleitung im Studium verlangt wird¹⁾.

2.2. Lernzielstufe 2: Modell anwenden

Während die Lernziele der Stufe 1 mit den beschriebenen Lehrmethoden relativ gut erreicht werden, ist es wesentlich schwieriger, Lehrstrategien zur Vermittlung von Anwendungswissen zu finden.

2.2.1. Übung

Am weitesten verbreitet sind Übungen auf dem Gebiet des Rechnungswesens, in denen die Studenten leichte Fälle lösen. Doch bereits in diesem mathematisch einfachen Stoffgebiet benötigen die Studenten sehr viel Zeit für Rechenoperationen. Soll in einer Übung dagegen ein Problem wie die Erstellung eines Investitionsbudgets behandelt werden, ergeben sich bereits große Probleme wegen des hohen Rechenaufwandes. Wenn aber beispielsweise in einer Sensitivitätsanalyse eines Investitionsbudgets untersucht werden soll, wie sich verschiedene Schätzungen des Absatzes auf die Qualität der Lösung auswirken, so wird der Umfang der mathematischen Operationen so groß, daß eine manuelle Durchrechnung unmöglich wird. Dieses Problem wird noch deutlicher, wenn man nun noch die einzelnen Schätzwerte, die der Investitionsentscheidung zugrunde liegen, durch stetige Verteilungen ersetzen und darauf eine Risikoanalyse aufbauen will. Der Einsatz der Übung (ohne Unterstützung des Rechners) ist somit auf nicht-formale und relativ einfache formale Modelle beschränkt.

1) Vgl. dazu Kosiol, E., Die Behandlung praktischer Fälle im betriebswirtschaftlichen Hochschulunterricht (Case Method), Berlin 1957, S. 12 ff.

2.2.2. Rechnerunterstützte Übung

Unter der Übungsstrategie im Rahmen des RUU wird meist das Training elementarer Fähigkeiten verstanden, was auch im anglo-amerikanischen Ausdruck "drill and practise" gut zum Ausdruck kommt. Das Üben mit komplexen Verfahren ist in den vorhandenen Unterrichtssystemen nicht vorgesehen und daher in der Übungsstrategie bisher nicht verwirklicht worden. Es gelten somit ähnliche Aussagen wie für die Übung ohne Rechner.

2.2.3. Fallstudie

Während in der Übung die Anwendung von Verfahren im Mittelpunkt steht, dient die Fallstudie mehr dem Erkennen von Entscheidungssituationen und der Untersuchung verschiedener Lösungsstrategien¹⁾. Dazu können natürlich auch mathematische Verfahren eingesetzt werden. Komplexere Modelle, wie sie von uns beispielhaft aufgezählt wurden, können aber auch hier aus denselben Gründen wie bei der Übung kaum oder überhaupt nicht verwendet werden. Die Fallstudie kann dem Studenten somit ebenfalls nur sehr beschränkt die Möglichkeit bieten, Anwendungserfahrung für die Verfahren der betrieblichen Entscheidungslehre zu sammeln, da sie keine Methoden einsetzen kann, die einen größeren Rechenaufwand erfordern.

2.2.4. Unternehmensplanspiel

Unter dem Begriff der Unternehmensplanspiele wurden in der letzten Zeit Modelle mit sehr unterschiedlichem Inhalt erstellt. Von der Simulation ganzer Industriezweige bis zum Planspiel, das Entscheidungen im Fertigungsbereich zum Gegenstand hat, wurde eine Reihe von Spielen mit unterschiedlichen Detailzielen entwickelt.

1) Vgl. dazu auch Steinmann, H., Kumar, B., Bleyer, E., Die Fallmethode in der universitären Führungsausbildung - Ergebnisse einer empirischen Befragung betriebswirtschaftlicher Lehrstühle, Universität Erlangen-Nürnberg, Arbeitspapier Nürnberg 1972, S. 17 ff.

Das "konventionelle" Planspiel (z.B. INTOP¹⁾, TOPIC 1²⁾, DELPHI³⁾ und OMNILOG III⁴⁾) stellt im Prinzip eine Dynamisierung der Fallstudie dar, wobei noch der für den Lerner (vor allem für die Motivation) interessante Aspekt hinzukommt, daß er das "Ergebnis" seiner Entscheidungen studieren kann.

Für die Anwendung betriebswirtschaftlicher Verfahren bringt diese Form des Planspiels im Vergleich zur Fallstudie keine neuen Vorteile.

Der Aspekt, daß der Manager seine Entscheidungen zunehmend durch formale Modelle unterstützen muß, ist bei folgenden Planspielen zum Teil berücksichtigt worden:

- IBGS

Unter dem Namen IBGS (Interactive Business Game Simulation)⁵⁾ hat Wahi ein dialogfähiges Planspiel erstellt und in APL programmiert, das ähnlich wie TOPIC 1 aufgebaut ist⁶⁾. Ein- und Ausgabe des Systems erfolgen jedoch on-line. Der Student hat

-
- 1) Vgl. Thorelli, H.B., Graves, R.L., Howells, L.T., Spielerhandbuch zum INTOP-Planspiel, Indiana University, Bloomington 1966.
Zum detaillierteren Studium von INTOP eignet sich besonders die Arbeit:
Amedick, W., Simulation des Unternehmerischen Entscheidungsprozesses - dargestellt am Beispiel eines komplexen Unternehmensspiels, Dissertation Berlin o.J., S. 42 ff.
 - 2) Vgl. dazu IBM (Hrsg.), IBM Unternehmensplanspiel TOPIC 1, Spielerhandbuch, IBM Form 71485-1.
Einen Überblick über TOPIC 1 gibt:
Faßhauer, R., Mit TOPIC 1 Top-Manager, absatzwirtschaft 14 (1971) 1, S. 17.
 - 3) Vgl. dazu Stubenrauch, H., DELPHI - ein handelsorientiertes Unternehmensspiel, data praxis, Bestell-Nr. 2-2600-829.
 - 4) Vgl. dazu Lange, D., Das Unternehmensplanspiel Honeywell Bull OMNILOG III, HB-Schriften 6.
 - 5) Vgl. dazu o.V., IBGS - Interactive Business Game Simulation, IBM Form GB 21-0393-0.
 - 6) Vgl. Wahi, P.N., A general management business simulation in APL, IBM Systems Journal 11 (1972) 2, S. 169 ff.

auch die Möglichkeit - und dies unterscheidet IBGS von den oben erwähnten Planspielen -, die vorhandenen Daten mit betriebswirtschaftlichen Verfahren (assisting tools) zu analysieren. Er kann beispielsweise mit Hilfe eines ebenfalls in APL programmierten Linearen Planungsmodells eine optimale Betriebsmittelzuteilung erstellen oder anhand der Regressionsanalyse Marktdaten untersuchen. IBGS stellt also über die Daten der Unternehmung hinaus noch Hilfsmittel zur Entscheidungsvorbereitung zur Verfügung.

- FDIC BMS

Wie IBGS läuft auch das Bankspiel FDIC BMS (Bank Management Simulation)¹⁾ im Dialog ab. Es unterscheidet sich neben anderen, hier nicht relevanten Punkten wie IBGS durch die Bereitstellung von Operations-Research-Verfahren von den "konventionellen" Planspielen. Der Student hat eine Menge von betriebswirtschaftlichen Instrumenten zur Verfügung, mit denen er in einem gewissen Umfang Analysen der vorhandenen Daten durchführen kann.

Diese Ansätze, den Manager auch in der Anwendung betriebswirtschaftlicher Verfahren auszubilden, erlauben es, die Lernzielstufe 2 für den von uns umrissenen Lehrstoff zu erreichen; dies ist allerdings auf die im Planspiel vorgesehenen Verfahren beschränkt.

Es ist nicht möglich, beliebige Methoden der Betriebswirtschaftslehre dazu aufzunehmen. Somit bieten Planspiele dieser Art nur partiell Hilfen für das hier untersuchte Ausbildungsproblem.

2.2.5. Problemlösung mit interaktiven Programmen

Als "Abfallprodukt" der betriebswirtschaftlichen Forschung und Praxis können Dialogsysteme für den Unterricht anfallen.

1) Hedberg, B., On Man-Computer Interaction in Organizational Decision-Making. A Behavioral Approach, Göteborg 1970.

So wurden am Betriebswirtschaftlichen Institut der Universität Erlangen-Nürnberg (Lehrstuhl für Betriebs- und Wirtschaftsinformatik) bereits mehrere Dialogprogramme zur Untersuchung betriebswirtschaftlicher Probleme erstellt. Ein Beispiel dafür ist das System INBESI (Investitionsbeurteilung mit Hilfe der Simulationstechnik)¹⁾, ein interaktives System, mit dessen Hilfe verschiedene Investitionsvorhaben beurteilt werden können. Es enthält die Teilsysteme INVEIN²⁾ (Investitionseinflußgrößenerfassung), INRISK³⁾ (Risikoanalyse zur Investitionsbeurteilung), INVOR⁴⁾ (Vorauswahl zur Investitionsbeurteilung), INSCOR⁴⁾ (Scoring-Methode zur Beurteilung von Investitionsalternativen) und INSALP⁵⁾ (Simulation alternativer Investitionsprogramme). Dieses System, das für praktische Probleme entwickelt wurde, kann nun nach einigen Modifikationen auch für den Unterricht eingesetzt werden. Die Vielzahl der Parameter und Entscheidungsvariablen vermittelt dem Studenten bei der Durcharbeitung eines Falles ein Problembewußtsein, das in dieser Art mit konventionellen Lehrmethoden nicht erreicht werden kann.

Ebenso wurde ein Verfahren zum Einsatz der Simulation bei der Break-even-Analyse in einem Dialogprogramm⁶⁾ erprobt. Dieses ist bereits für den Unterricht bereitgestellt worden.

-
- 1) Emmert, P., Investitionsbeurteilung im Wege der Mensch-Maschine-Kommunikation, Dissertation (in Arbeit) Nürnberg.
 - 2) Schneider, H., Beurteilung qualitativer Einflußgrößen bei Investitionsobjekten in einem EDV-Dialogmodell mit Berücksichtigung einer durch quantitative Kriterien getroffenen Vorauswahl, Diplomarbeit Nürnberg 1972.
 - 3) Schießl, W., Risikoanalyse von Investitionsvorhaben im Rahmen eines EDV-Dialog-Modells, Diplomarbeit Nürnberg 1972.
 - 4) Kroll, W., Aufstellen alternativer Investitionsprogramme mit Berücksichtigung des Etats in einem EDV-Dialog-Verfahren, Diplomarbeit Nürnberg 1972.
 - 5) Stumpf, R., Erfassung und Zusammenspiel quantitativer Einflußgrößen bei Investitionsvorhaben in einem computerunterstützten Simulationsmodell, Diplomarbeit Nürnberg 1972.
 - 6) Rackelmann, G., a.a.O.

An der Sloan School des MIT wurde das System IGRAM (Interactive Graphical Risk Analysis Method)¹⁾ entwickelt. Das Programm liegt bereits in der vierten Version vor und hat nun den Namen "Hierarchical Modelling System" (HMS). Es handelt sich um ein Entscheidungsbaumverfahren, in das der Student Werte auf deterministischer, probabilistischer und "abhängiger" Basis eingeben kann. Das Programm berechnet die Knoten höherer Ordnung in einem Entscheidungsbaum. Rockart schreibt dazu sinngemäß²⁾:

In diesem Modell müssen möglichst viele Elemente eines hierarchischen Entscheidungsbaumes quantifiziert und die Verteilung möglicher Ergebnisse vor dem Treffen der Entscheidung betrachtet werden. Wir versuchen gerade, diesen Prozeß unseren Schülern an der Sloan School zu lehren. Die Studentenversuche haben gezeigt, daß die Methode in zweifacher Weise nützlich ist: bei der Strukturierung und der Lösung von Problemen.

Die hier beschriebenen Programme erlauben es dem Studenten, Verfahren anzuwenden und deren Eigenschaften zu studieren, wie es mit anderen Unterrichtsmethoden nicht möglich ist. Sie sind allerdings nicht für den Unterricht geschaffen worden und bleiben zudem solange isolierte Einzelentwicklungen, bis ein Unterrichtssystem geschaffen ist, das diese Kursteile aufnehmen kann und das vor allem eine weitere geplante Entwicklung eines Curriculums erlaubt.

2.3. Lernzielstufe 3: Modell modifizieren

Für den Betriebswirt genügt es in der Praxis meist, vorhandene Verfahren des Operations Research und der Datenverarbeitung anwenden oder deren Ergebnisse interpretieren zu können.

1) Vgl. Rockart, J.F., a.a.O., S. 39 ff.

2) Ebenda, S. 39.

Für den Forscher ist es aber unumgänglich notwendig, ein Verfahren vollständig zu verstehen, d.h. zu analysieren und in der Lage zu sein, selbst Veränderungen durchzuführen oder neue Verfahren zu konstruieren. Er soll auch imstande sein, verschiedene Wege einer Problemlösung kritisch zu beurteilen.

Seminar und Kolloquium sind zwei Lehrveranstaltungen, in denen der Lernende unter Anleitung des Dozenten Modelle modifizieren und beurteilen lernt. Am besten kann der Student aber diese Lernzielstufe erreichen, wenn er betriebswirtschaftliche Rechnerprogramme erstellt, da die formale Darstellung ein gründliches Verständnis des Verfahrens voraussetzt.

3. Zusammenfassung und Schlußfolgerung:

Lücken im vorhandenen Angebot von Lehrmethoden

Die Lernziele der Stufe 1 werden mit den vorhandenen Lehrmethoden hinreichend erfüllt.

Obwohl die Lernzielstufe 2 für eine praxisorientierte betriebswirtschaftliche Ausbildung besonders wichtig ist, reichen die aufgeführten Lehrmethoden nicht aus, diese im notwendigen Umfang zu ermitteln.

Nur wenige spezielle Unternehmensplanspiele und verschiedene interaktive Programme sind zur Vermittlung des entsprechenden Wissens vorhanden. Diese sind aber nicht zur Lehre der Verfahrensanwendung geschaffen worden und weisen deshalb Mängel auf.

Es fehlt ein Lehrverfahren, das es dem Studenten erlaubt, Verfahren, die er theoretisch bereits durchdrungen hat, auf praxisnahe Fälle anzuwenden, und das somit dem Betriebswirt jene Möglichkeit bieten kann, die für den Naturwissenschaftler in Form des Praktikums selbstverständlich ist.

Z w e i t e r T e i l

Konzeption der Methode CET

=====

Die Lehrmethode "Computerunterstütztes Entscheidungstraining" (CET) soll helfen, die in Teil 1, Punkt 3, beschriebene Lücke der Ausbildung in der betriebswirtschaftlichen Entscheidungslehre zu beseitigen.

1. Einführendes Beispiel: Darstellung einer Terminalsitzung

Um einen ersten Überblick über die Lehrmethode des CET zu verschaffen, wird anschließend eine hypothetische Terminalsitzung eines Studenten dargestellt, die zwar nicht typisch für solche Sitzungen ist, zur Darstellung des CET-Gedankens aber gut geeignet sein dürfte. Ähnliche Passagen wie im Beispiel sind bereits im Dialogsystem des CET realisiert worden¹⁾.

Da es aber das Ziel dieser Arbeit ist, Anforderungen zur Verwirklichung des CET zu entwerfen, richtet sich dieses Beispiel primär nach den didaktischen Erfordernissen. Es ist mit vertretbarem Aufwand in den vorhandenen Softwaresystemen nicht realisierbar, stellt aber keine Anforderungen, die in den nächsten Jahren nicht softwaremäßig verwirklicht werden können.

Zum Verständnis des Beispiels muß noch gesagt werden, daß die Studentenantworten stets unterstrichen sind. Der Student erhält am Terminal zuerst ein Bild, arbeitet dieses durch und schreibt dann seine Eingabe in die erste Zeile desselben Bildes. Beim Lesen des Beispiels muß auf diese Eigenart der Bildschirmkommunikation geachtet werden.

1) Vgl. dazu Reitbauer, F., Das Trainingssystem im Computerunterstützten Entscheidungstraining, Dissertation (in Arbeit) Nürnberg.

Teile der Darstellung des Beispiels wurden als Hardcopies von Bildschirminhalten des bestehenden Systems übernommen.

Ein CET-Ablauf bei der Lehre der exponentiellen Glättung

Dialog:

Bild 1:

```
*****
*STUDENTIX, 671563
*GEBEN SIE BITTE IHREN NAMEN UND IHRE NUMMER EIN.
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*****
```

Erklärung:

Zur Identifikation und Sicherung des schülereigenen Datenbereichs werden Schülername und -nummer eingegeben.

Bild 2:

```
*****
*1
*
*WOLLEN SIE
* 1 LERNEN ODER
* 2 PROBLEMLÖSEN?
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*****
```

Wenn der Student das CET-System zur Lösung von Problemen benutzen will, wird im Dialog anders fortgefahren als beim Lernen.

Bild 3:

```
*****
*4
*AN WELCHEM DER FOLGENDEN KURSE WOLLEN SIE ARBEITEN?
*
* 1 LAGERHALTUNG
* 2 PRODUKTIONSPLANUNG
* 3 INVESTITION
* 4 PROGNOSERECHNUNG
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*****
```

Da der Student gleichzeitig an mehreren Kursen arbeiten kann, muß gefragt werden, welchen Kurs er in der konkreten Situation will. Es werden dazu all jene Kurse ausgegeben, an denen der Student gerade arbeitet.

Bild 4:

```
*****
*
*
*SIE HABEN IN IHRER LETZTEN SITZUNG FOLGENDE FRA-
*GE(N) AN DEN LEHRER GESTELLT:
*
*WIESO IST DAS VERFAHREN VOM WINTERS OHNE SAISON-
*KOMPONENTE NICHT VORHANDEN?
*
*ANTWORT:
*DIE BESONDERHEIT DES WINTERS-VERFAHRENS IST, DASS
*ES FÜR SAISONALE ABLÄUFE KONZIPIERT WURDE. EIN
*ANDERES WINTERS-VERFAHREN GIBT ES NICHT.
*
*
*
*
*
*
*****
```

Der Schüler kann mit dem Befehl *FRAGE einen Text für den Lehrer eingeben. Dieser kann alle Fragen in bestimmten Perioden beantworten. Der Schüler bekommt somit auch Informationen, die über die im Kurs fest vorgesehenen hinausgehen.

Bild 5:

```
*****
*
*
*SIE HABEN FÜR DIESEN KURS BISHER 03 MIN. 36 SEC.
*RECHENZEIT VERBRAUCHT.
*ES STEHT IHNEN NOCH RECHENZEIT IM UMFANG VON
*06 MIN. 24 SEC. ZUR VERFÜGUNG.
*
*
*
*
*
*
*
*
*
*
*
*
*****
```

Der Student hat eine bestimmte Menge Rechenzeit, die vom Kursautor festgelegt und vom Lehrer modifiziert werden kann, zur Verfügung. Er kann sie nach seiner eigenen Wahl für die einzelnen Kursteile verwenden. Die Berechnung und Ausgabe wird im Rahmen der Lernersteuerung (LS) durchgeführt.

Bild 6:

```
*****
*1
*
*SIE HABEN FOLGENDE MOEGlichkeiten ZUR FORTSETZUNG
*IHRES KURSES:
*
* 1 RF3 TEST UEBER EXPONENTIELLE GLAETUNG
* 2 RF3 TEST UEBER SCHRITTWEISE REGRESSION MIT
*   ADAPTIVER EINFLUSSGROESSENKOMBINATION
* 3 RF2 WIEDERHOLUNG DER DATENANALYSE
* 4 RF9 TRAINING DER EXPONENTIELLEN GLAETUNG
* 5 RF1 EMPFEHLUNGEN FÜR DAS LITERATURSTUDIUM
* 6   ANDERE ALTERNATIVE
*
*
*
*
*
*
*****
```

Die LS erstellt für den Studenten aufgrund seiner Lern-daten eine Reihe von Vorschlä- gen, die nach absteigender Eignung geordnet sind. Der Student kann auch eine nicht vorgesehene Alternative wäh- len, wenn er (hier mit Nummer 6) eine andere Fortsetzung an- gibt.

RF (Rechenzeitfaktor) gibt dem Studenten an, wie rechen- intensiv eine Methode ist.

Bild 7:

```
*****
*105
*
*BERECHNEN SIE BITTE DEN WERT SX (T + X) MIT HILFE
*DES KONSTANTEN MODELLS MIT EXPONENTIELLER GLAET-
*TUNG, WENN FOLGENDE WERTE GEGEBEN SIND (ES MUES-
*SEN NICHT ALLE WERTE VERWENDET WERDEN):
*
*T      {INDEX DER GLAETTUNGSPERIODE} = 1
*K      {ANZAHL DER ZU PROGNOSTIZIERENDEN PERIODEN}=1
*X(1)   {BEOBACHTUNGSWERT ZUM ZEITPUNKT 1} = 100
*X(2)   {BEOBACHTUNGSWERT ZUM ZEITPUNKT 2} = 110
*A      {GRUNDWERT IN DER PERIODE 0} = 100
*B      {TRENDWERT IN DER PERIODE 0} = 5
*C      {TRENDWERT DER TRENDWERTE IN DER PERIODE 0}=
*      0.003
*ALFA   {GLAETTUNGSPARAMETER} = 0.5
*
*GEBEN SIE BITTE IHR ERGEBNIS EIN!
*
*****
```

Der Student hat die laut Vor-
schlag des Rechners beste
Alternative gewählt und er-
hält sofort die Fragen zum
entsprechenden Gebiet. Wäh-
rend des Tests können keine
anderen CET-Strategien be-
nutzt werden, damit die Test-
ergebnisse nicht verfälscht
werden.

Bild 8:

```
*****
*2
*
*RICHTIG
*WANN WERDEN DIE NEUEREN BEOBACHTUNGSWERTE STAERKER
*BERUECKSICHTIGT:
*
* 1 WENN ALFA = 0.3
* 2 WENN ALFA = 0.4
*
*GEBEN SIE BITTE 1 ODER 2 EIN.
*
*
*
*
*
*
*
*
*
*
*
*****
```

Nach Beantwortung einer Frage
bekommt der Student sofort
eine Meldung, ob seine Einga-
be richtig oder falsch war.

Bild 9:

```
*****
*4
*
*RICHTIG
*WELCHE DER FOLGENDEN AUSSAGEN IST RICHTIG?
*DER GLAETTUNGSPARAMETER ALFA LIEGT IN DER PRAXIS
*GEWOEHNLICH
*
* 1 ZWISCHEN 0.3 UND 0.5
* 2 ZWISCHEN 0.1 UND 0.3
* 3 ZWISCHEN 0.1 UND 0.7
* 4 ZWISCHEN 0.7 UND 0.9
*
*
*
*
*
*
*
*
*
*
*
*****
```

Bild 10:

```
*****
*15.7*
*FALSCH*
*ES LIEGT FOLGENDE ZAHLENREIHE VOR:*
*
* $X(1) = 14$       $X(6) = 17$ *
* $X(2) = 14$       $X(7) = 15$ *
* $X(3) = 16$       $X(8) = 16$ *
* $X(4) = 17$       $X(9) = 15$ *
* $X(5) = 18$       $X(10) = 15$ *
*
*GEBEN SIE DEN GRUNDWERT DIESER ZAHLENREIHE AN.*
*
*
*
*
*
*
*
*****
```

Bild 11:

```
*****
*2*
*
*RICHTIG*
*WELCHE DER FOLGENDEN AUSSAGEN STIMMT?*
*
* 1 WIRD EIN KONSTANTES MODELL AUF EINE ZEITREIHE*
* MIT POSITIVEM TREND ANGEWANDT, SO LIEGT DIE*
* SCHAEZTUNG FUER X IM DURCHSCHNITT IMMER UEBER*
* DEM TATSAECHLICHEN WERT VON X.*
*
* 2 WIRD EIN KONSTANTES MODELL AUF EINE ZEITREIHE*
* MIT POSITIVEM TREND ANGEWANDT, SO LIEGT DIE*
* SCHAEZTUNG FUER X IM DURCHSCHNITT IMMER UNTER*
* DEM TATSAECHLICHEN WERT VON X.*
*
* 3 WIRD EIN KONSTANTES MODELL AUF EINE ZEITREIHE*
* MIT POSITIVEM TREND ANGEWANDT, SO FALLEN DIE*
* SCHAEZTUNGEN FUER X UND DIE TATSAECHLICHEN*
* BEOBACHTUNGSWERTE IMMER ZUSAMMEN, WENN KEINE*
* ZUFALLSABWEICHUNGEN VORHANDEN SIND.*
*****
```

Bild 12:

```
*****
*1*
*
*RICHTIG*
*WELCHE DER FOLGENDEN AUSSAGEN STIMMT?*
*
*WIRD EIN KONSTANTES MODELL AUF EINE ZEITREIHE MIT*
*SAISONALEM VERLAUF, BEI DER KEINE ZUFALLSSCHWAN-  
*KUNGEN VORHANDEN SIND, ANGEWANDT,*
* 1 SO VERLAEUFT DIE KURVE DER SCHAEZTWERTE STETS*
* FLACHER ALS DIE KURVE DER BEOBACHTUNGSWERTE,*
* 2 SO VERLAEUFT DIE KURVE DER BEOBACHTUNGSWERTE*
* STETS FLACHER ALS DIE KURVE DER SCHAEZTWERTE.*
* 3 SO SIND DIE SCHAEZTWERTE STETS GROESSER ALS*
* DIE BEOBACHTUNGSWERTE.*
*
*
*
*
*
*
*
*****
```


Bild 16:

```
*****
*
*
*EIN HOHES ALFA LAESST DAHER DAS MODELL SEHR
*SCHNELL AUF VERAENDERUNGEN IN DER ZEITREIHE REA-
*GIEREN! ES WIRD SEHR ANPASSUNGSFAEHIG, DADURCH
*ABER AUCH INSTABIL.
*EIN NIEDRIGES ALFA MACHT DAS MODELL STABIL. ES
*WIRD ABER WENIGER ANPASSUNGSFAEHIG.
*DIE IN DER PRAXIS VERWENDETEN WERTE FUER ALFA BE--
*WEGEN SICH MEIST ZWISCHEN 0.1 UND 0.3. DIESE GENE--
*RELLE AUSSAGE IST ABER NUR SEHR GROB.
*
*
*
*
*
*
*
*
*
*
*****
```

Bild 17:

```
*****
*2
*
*WOLLEN SIE
* 1 NOCH ANDERE TEILE WIEDERHOLEN ODER
* 2 DAS WIEDERHOLUNGSPROGRAMM BEENDEN?
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*****
```

Bild 18:

```
*****
*3
*
*SIE HABEN FOLGENDE MOEGlichkeiten ZUR FORTSETZUNG
*IHRES KURSES:
*
* 1 RF3 TEXT UEBER SCHRITTWEISE REGRESSION MIT
*   ADAPTIVER EINFLUSSGROESSENKOMBINATION
* 2 RF2 WIEDERHOLUNG DER DATENANALYSE
* 3 RF9 TRAINING DER EXPONENTIELLEN GLAETTUNG
* 4 RF1 EMPFEHLUNGEN FUER DAS LITERATURSTUDIUM
* 5   ANDERE ALTERNATIVEN
*
*
*
*
*
*
*
*
*
*
*****
```

Der Student bekommt die nach dem Text für exponentielles Glätten erster Ordnung revidierten Vorschläge der LS.

Bild 19:

```
*****
**MERKE PROBLEM**
**
**SIE HABEN FOLGENDE 24 VERGANGENHEITSWERTE (MONATS-
**UMSAETZE DES ARTIKELS "A 412") UND SOLLEN NUN DEN
**UMSATZ FUER DIE NAECHSTEN DREI MONATE PROGNOSTI-
**ZIEREN.
**
**MONAT    JAHR    UMSATZ    MONAT    JAHR    UMSATZ
**OKTOBER  1970    222.3    OKTOBER  1971    234.7
**NOVEMBER 1970    220.7    NOVEMBER 1971    234.0
**DEZEMBER 1970    219.1    DEZEMBER 1971    233.2
**JANUAR   1971    219.4    JANUAR   1972    235.2
**FEBRUAR  1971    219.4    FEBRUAR  1972    238.3
**MAERZ    1971    224.0    MAERZ    1972    241.0
**APRIL    1971    231.8    APRIL    1972    244.8
**MAI      1971    235.7    MAI      1972    249.9
**JUNI     1971    237.2    JUNI     1972    251.8
**JULI     1971    238.5    JULI     1972    251.7
**AUGUST   1971    238.1    AUGUST   1972    253.5
**SEPTEMBER 1971    236.9    SEPTEMBER 1972    249.3
*****
```

Der Student erhält hier seine Problemstellung. Er gibt den Befehl: *MERKE PROBLEM. Dies bewirkt, daß er den Inhalt von Bild 8 stets durch den Befehl *ZEIGE PROBLEM wieder aufrufen kann.

Bild 20:

```
*****
*1
**
**IM RAHMEN JEDER TRAININGSLEKTION DES KURSES PRO-
**GNOSERECHNUNG STEHEN IHNEN FOLGENDE VERFAHRENS-
**GRUPPEN ZUR VERFUEGUNG:
**
** 1 DATENANALYSE
** 2 MODELLBILDUNG
** 3 PROGNOSE
** 4 BEURTEILUNG
**
**WAEHLN SIE BITTE EINE DER ALTERNATIVEN DURCH EIN-
**GABE DER ENTSPRECHENDEN NUMMER!
**
**
**
**
**
**
*****
```

Der Student hat die Möglichkeit, aus vier Gruppen von Verfahren zur Unterstützung der Teiltätigkeiten einer Prognoseerstellung zu wählen.

Bild 21:

```
*****
**GRA
**
**WELCHES DER FOLGENDEN VERFAHREN WOLLEN SIE ZUR
**ANALYSE IHRER DATEN VERWENDEN?
**
** *GRA: GRAPHISCHE METHODE
** *D1 : ERSTE DIFFERENZEN
** *D2 : ZWEITE DIFFERENZEN
** *D3 : DRITTE DIFFERENZEN
** *QUO: QUOTIENTENBILDUNG
** *GDU: GLEITENDE DURCHSCHNITTE
** *KOR: KORRELATION MIT DER ZEIT
**
**
**
**
**
**
*****
```

Der Student hat für die Datenanalyse neben den generell vorhandenen Befehlen noch solche, mit denen er einzelne Verfahren ansprechen kann.

Bild 22:

```

*****
**WERKE GRAPHIK *BEFEHLE**
**
** 253I
** 250I
** 247I
** 244I
** 241I
** 238I
** 235I
** 232I
** 229I
** 226I
** 223I*
** 220I ****
** 217I
** 214I
** 211I
** 208I
** 205I
** 202I-----
**      5      10      15      20
*****

```

Der Student erhält nach Eingabe des Befehls *GRA die Graphik der Beobachtungsdaten. Er speichert den Inhalt von Bild 11 unter "GRAPHIK" ab.

Bild 23:

```

*****
**D1
**
**ES STEHEN IHNEN FOLGENDE VERFAHREN ZUR VERFUEGUNG:**
**
** *GRA: GRAPHISCHE METHODE
** *D1 : ERSTE DIFFERENZEN
** *D2 : ZWEITE DIFFERENZEN
** *D3 : DRITTE DIFFERENZEN
** *QUO: QUOTIENTENBILDUNG
** *QDU: GLEITENDE DURCHSCHNITTE
** *KOR: KORRELATION MIT DER ZEIT
**
**
**
**
**
**
**
**
**
**
*****

```

Durch den Befehl *BEFEHLE in Bild 11 erhält der Student die Auflistung der im Rahmen der Datenanalyse möglichen Befehle. Bei nochmaligem Eingeben von *BEFEHLE würden die generell möglichen Befehle aufgelistet.

Bild 24:

```

*****
**D2
**
**DIE ERSTEN DIFFERENZEN BETRAGEN:
**
**  -1.6   -1.5     .2     .0     4.6     7.3
**   3.8    1.5    1.3    -1.3   -1.2   -2.1
**   -6     -8     2.0    3.1    2.6    3.8
**    5.0    1.9    -0.1    1.8   -4.1
**
**
**
**
**SUMME DER ERSTEN DIFF.=      27.0
**
**
**
**
*****

```

Bild 25:

```
*****
**RETURN
**
**DIE ZWEITEN DIFFERENZEN BETRAGEN#
**
**      .J      1.3      -0.2      4.6      3.1      -3.9
**    -2.3      -0.1      -1.6      -0.8      -0.9      1.5
**      -0.1      2.8      1.1      -0.4      1.1      1.2
**    -3.1      -2.0      1.9      -6.6
**
**
**
**SUMME DER ZWEITEN DIFF.=      -2.5
**
**
**
**
**
**
*****
```

Durch den Befehl *RETURN gelangt der Student an jenen Punkt zurück, von dem er die Datenanalyse aufgerufen hat.

Bild 26:

```
*****
3
**
**
**WELCHE DER FOLGENDEN VERFAHRENSGRUPPEN WOLLEN SIE
**NUN WAEHLEN?
**
** 1 DATENANALYSE
** 2 MODELLBILDUNG
** 3 PROGNOSE
** 4 BEURTEILUNG
**
**
**
**
**
**
**
**
**
**
**
*****
```

Bild 27:

```
*****
**
**
**ZUM TRAINING DER EXPONENTIELLEN GLAETTUNG STEHEN
**IHNNEN FOLGENDE BEFEHLE ZU DEN VERFAHREN ZUR VER-
**FUEGUNG:
**
** *EG1 DIESER BEFEHL BEWIRKT DIE BERECHNUNG DER
**      EXPONENTIELLEN GLAETTUNG ERSTER ORDNUNG MIT*
**      ANSCHLIESSENDE AUSGABE DER ERGEBNISSE IN
**      EINER GRAPHIK UND EINER TABELLE.
**
** *EG2 WIE EG1 FÜR EXPO. GLAETTUNG ZWEITER ORDNUNG*
**
** *EG3 WIE EG1 FÜR EXPO. GLAETTUNG DRITTER ORDNUNG*
**
**FORTSETZUNG DER BEFEHLSLISTE IM NAECHSTEN BILD.
**
**
**
*****
```

Der Autor hat die Möglichkeit, dem Studenten neben den allgemeinen Befehlen des Systems und den kurzspezifischen Befehlen auch solche zur Verfügung zu stellen, die nur für ein einziges Verfahren (hier exponentielles Glätten) Gültigkeit haben. Diese werden ihm erklärt, wenn er zum ersten Mal in eine bestimmte Trainingslektion kommt. Die Erklärung der Befehle könnte auch in einem Kursmanual stehen.

Bild 28:

```
*****
**LEX EG1
**
**FORTSETZUNG DER BEFEHLSLISTE:
**
** *ALFA=: MIT DEM BEFEHL "ALFA=" KOENNEN SIE DEN
**          GLAETTUNGSPARAMETER VERAENDERN
**          (Z.B.: *AL = 0.3).
**
** *A=: WIE ALFA KOENNEN SIE DEN GLAETTUNGS-
**        WERT A DER AUSGANGSPERIODE MODIFIZIEREN.*
**
** *B=: TRENDWERT DER AUSGANGSPERIODE
**
** *C=: TRENDWERT DES TRENDWERTES DER AUSGANGS-
**        PERIODE
**
** *TANF=: PERIODE, AB DER GEGLAETTET WERDEN SOLL.
**
** *TEND=: PERIODE, BIS ZU DER GEGLAETTET WERDEN
**          SOLL (INKLUSIV).
*****
```

Mit Hilfe von "Zuweisungsbe-
fehlen" kann der Student Wer-
te für Parameter eingeben.

Mit der Eingabe *LEX EG1 ver-
langt der Student vom Lexikon
eine Beschreibung des Modells
der exponentiellen Glättung
erster Ordnung. Es sind zu
jedem Modell unter dem in der
Befehlsliste angegebenen Na-
men Erklärungen im Lexikon
gespeichert.

Bild 29:

```
*****
**ZEIGE PROBLEM
**
**BEIM KONSTANTEN MODELL MIT EXPONENTIELLER GLAET-
**TUNG (EXPONENTIELLE GLAETUNG ERSTER ORDNUNG)
**HABEN SIE ES MIT FOLGENDEN MATHEMATISCHEN MODELL
**ZU TUN:
**
** SX (T+K) = A
** A = ALFA * X(T) + (1-ALFA) * SX(T)
** SX (T+K)... SCHAETZUNG DER VARIABLEN X FÜR DEN
**              ZEITPUNKT T+K
** X(T)..... WERT DER VARIABLEN X ZUM ZEITPUNKT T
** ALFA..... GLAETTUNGSPARAMETER
** A..... GRUNDWERT
** WEITERE BEGRIFFE:
** EXPONENTIELLES GLAETTEN
** GLAETTUNGSPARAMETER
** GRUNDWERT
**
*****
```

Diese Kurzerklärung des Mo-
dells wird durch den Befehl
*LEX EG1 aufgerufen. Will der
Student mehr wissen, muß er
Befehle wie *LEX oder *WIEDER-
HOLUNG (ein globaler Lexikon-
aufruf) eingeben.
Der Student will vor der Ein-
gabe der Parameter die Pro-
blemstellung noch einmal
sehen.

Bild 30:

```
*****
**ZEIGE GRAPHIK
**
**SIE HABEN FOLGENDE 24 VERGANGENHEITSWERTE (MONATS-
**UMSAETZE DES ARTIKELS "A 412") UND SOLLEN NUN DEN
**UMSATZ FÜR DIE NÄCHSTEN DREI MONATE PROGNOSTI-
**ZIEREN.
**
**MONAT   JAHR   UMSATZ   MONAT   JAHR   UMSATZ
**OKTOBER 1970   222.3   OKTOBER 1971   234.7
**NOVEMBER 1970   220.7   NOVEMBER 1971   234.0
**DEZEMBER 1970   219.1   DEZEMBER 1971   233.2
**JANUAR   1971   219.4   JANUAR   1972   235.2
**FEBRUAR  1971   219.4   FEBRUAR  1972   238.3
**MÄRZ     1971   224.0   MÄRZ     1972   241.0
**APRIL    1971   231.8   APRIL    1972   244.8
**MAI      1971   235.7   MAI      1972   249.9
**JUNI     1971   237.2   JUNI     1972   251.8
**JULI     1971   233.5   JULI     1972   251.7
**AUGUST   1971   238.1   AUGUST   1972   253.5
**SEPTEMBER 1971  236.9   SEPTEMBER 1972  249.3
*****
```

Er verlangt zur Veranschau-
lichung noch einmal die Gra-
phik der Beobachtungswerte.

Bild 34:

```

*****
*
*   EXPONENTIELLE GLAETTUNG ERSTER ORDNUNG*
*VON PER. 1 BIS PER. 11   ALFA = .30
*M.A.ABW= 6.58
*
*PERIODE      BEOBACHT.GRUNDWT.PROGNOSE ABS.ABW.*
* 1          222.30 230.70 230.00 7.69 *
* 2          220.70 227.69 227.69 6.98 *
* 3          219.10 225.53 225.53 6.49 *
* 4          219.40 223.64 223.64 4.24 *
* 5          219.40 222.37 222.37 2.97 *
* 6          224.00 221.48 221.48 2.51 *
* 7          231.30 222.23 222.23 9.56 *
* 8          235.70 225.10 225.10 10.59 *
* 9          237.20 228.28 228.28 8.91 *
* 10         238.50 231.95 231.95 7.54 *
* 11         239.10 233.22 233.22 4.87 *
* 12         236.50 234.63
* 13         234.70
* 14         234.00
*****

**A = 222 *ALFA = 0.5 *EG1
*
*   EXPONENTIELLE GLAETTUNG ERSTER ORDNUNG*
*VON PER. 1 BIS PER. 11   ALFA = .30
*M.A.ABW= 6.58
*
*PERIODE      BEOBACHT.GRUNDWT.PROGNOSE ABS.ABW.*
* 15         233.20
* 16         235.20
* 17         238.30
* 18         241.00
* 19         244.80
* 20         249.50
* 21         251.80
* 22         251.70
* 23         253.50
* 24         249.30
*
*
*****

```

Bild 35:

```

*****
**TEND = 15 *EG1
*
* 253I
* 250I
* 247I
* 244I
* 241I
* 238I
* 235I
* 232I
* 229I
* 226I
* 223I*+
* 220I*+000++
* 217I
* 214I
* 211I
* 208I
* 205I
* 202I-----
*
* 5 10 15 20
*****

```

Der Student erhöht die Anzahl der Glättungsperioden, um zu sehen, wie sich die Parametereinstellung für einen weiteren Teil der Zeitreihe bewährt.

Bild 36:

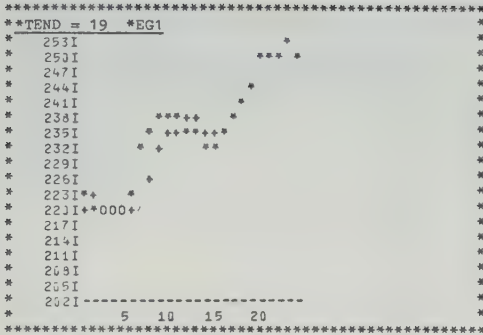
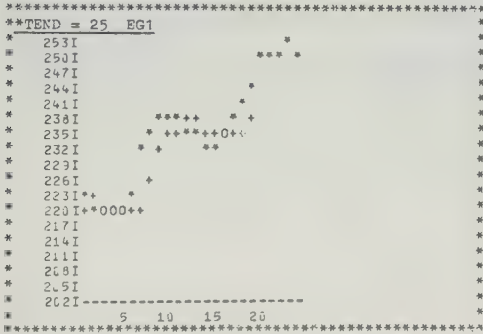


Bild 37:



Der Student ist nun zufrieden
und erstellt eine Prognose
für eine Periode.

Bild 38:

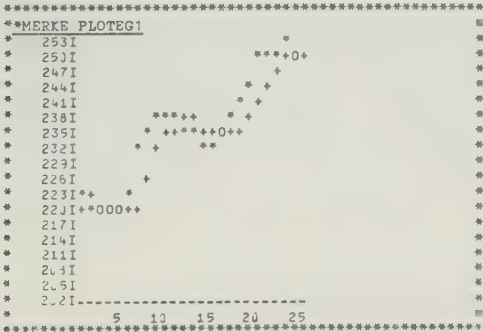


Bild 39:

```
*****
* EXPONENTIELLE GLAETUNG ERSTER ORDNUNG:
* VON PER. 1 BIS PER. 25 ALFA = .50
* M.A.ABW= 3.54
*
* PERIODE      BEOBACHT.GRUNDWT.PROGNOSE ABS.ABW.
* 1            222.30 222.00 222.00 0.30
* 2            224.70 222.15 222.15 1.45
* 3            219.10 221.42 221.42 2.32
* 4            219.40 221.26 221.26 0.86
* 5            219.40 219.33 219.33 0.43
* 6            224.00 219.61 219.61 4.39
* 7            231.80 221.80 221.80 9.99
* 8            239.70 226.30 226.30 8.89
* 9            237.20 231.25 231.25 5.94
* 10           236.50 234.22 234.22 4.27
* 11           236.10 235.36 235.36 1.73
* 12           236.90 237.23 237.23 0.33
* 13           234.70 237.06 237.06 2.36
* 14           234.00 235.38 235.38 1.38
* 15           233.20 234.94 234.94 1.74
*****
```

```
*****
* LEX EG2
*
* EXPONENTIELLE GLAETUNG ERSTER ORDNUNG:
* VON PER. 1 BIS PER. 25 ALFA = .50
* M.A.ABW= 3.54
*
* PERIODE      BEOBACHT.GRUNDWT.PROGNOSE ABS.ABW.
* 16           235.20 234.07 234.07 1.12
* 17           238.30 234.63 234.63 3.66
* 18           241.00 236.46 236.46 4.53
* 19           244.80 238.73 238.73 6.05
* 20           249.50 241.76 241.76 7.73
* 21           251.30 247.33 247.33 3.95
* 22           251.70 248.61 248.61 3.08
* 23           253.50 251.25 251.25 2.24
* 24           249.30 251.87 251.87 2.57
* 25           251.50 251.50 251.50 0.00
*
*
*
*****
```

Bild 40:

```
*****
* LEX TRENDWERT
*
* BEIM LINEAREN MODELL MIT EXPONENTIELLER GLAETUNG
* (EXPONENTIELLE GLAETUNG ZWEITER ORDNUNG) HABEN
* SIE ES MIT FOLGENDEM MATHEMATISCHEN MODELL ZU TUN:
*  $SX(T+K) = A + B * K$ 
*  $A = X(T) + (1 - ALFA) * 2 * (SX(T) - X(T))$ 
*  $B = B - ALFA * 2 * (SX(T) - X(T))$ 
*  $SX(T+K)...$  SCHAETZUNG DER VARIABLEN X FÜR DEN
* ZEITPUNKT T+K
*  $X(T).....$  BEOBACHTUNGSWERTE DER VARIABLEN X ZUM
* ZEITPUNKT T
* A..... GRUNDWERT
* B..... TRENDWERT
* ALFA..... GLAETUNGSPARAMETER
* WEITERE BEGRIFFE:
* EXPONENTIELLES GLAETEN
* GLAETUNGSPARAMETER
* GRUNDWERT
* TRENDWERT
*****
```

Der Student ist sich über die Bedeutung des Trendwertes nicht sicher und geht daher ins Lexikon.

Bild 41:

[illegible]

Der Student erhält die Erklärung jenes Begriffes, der laut einer syntaktischen Analyse am ehesten dem Suchbegriff entspricht. Es werden ihm gleichzeitig noch Begriffe ausgegeben, die eine Ähnlichkeit mit dem Suchbegriff haben.

Bild 42:

[illegible]

Da der Student in Bild 41 keine Befehle eingegeben hat, wird er vom Programm weitergesteuert. Der Student kann aber durch Eingabe von Befehlen jederzeit auf den Ablauf Einfluß nehmen.

Bild 43:

Scatter plot showing the number of publications (n) versus the year of publication (a) for the topic "Epidemiology of infectious diseases". The x-axis represents the year (a) from 1970 to 2010, and the y-axis represents the number of publications (n) from 0 to 250. The plot shows a general upward trend in the number of publications over time, with a significant increase starting around 1990. A dashed horizontal line is drawn at n=200.

Year (a)	Number of Publications (n)
1970	253
1971	250
1972	247
1973	244
1974	241
1975	238
1976	235
1977	232
1978	229
1979	226
1980	223
1981	221
1982	217
1983	214
1984	211
1985	208
1986	205
1987	202

Bild 44:

```
*****
*      EXPONENTIELLE GLAETTING ZWEITER ORDNUNG      *
*VON PER. 1 BIS PER. 11 ALFA = .60
*M.A.A3W= 2.64
*
*PERIODE BEOBACHT.GRUNDWT.TRENDWT.PROGNOSE ABS.ABW.*
* 1 222.33 221.93 .83 222.76 2.28 *
* 2 221.73 221.93 .83 222.76 2.66 *
* 3 219.13 221.13 .33 221.12 2.02 *
* 4 219.40 219.42 -.63 218.78 .61 *
* 5 219.40 219.30 -.41 218.88 .51 *
* 6 221.10 219.31 -.23 219.08 4.91 *
* 7 231.33 227.21 1.93 224.75 7.14 *
* 8 235.73 230.67 4.07 234.74 .95 *
* 9 237.23 235.54 4.41 233.96 2.76 *
* 10 238.53 237.64 3.42 241.16 2.56 *
* 11 238.13 238.91 2.49 241.43 3.33 *
* 12 236.73 238.62 1.30 *
* 13 234.73 *
* 14 234.33 *
*****
```

Da der Student nicht weiß,
wie es jetzt weitergehen
soll, gibt er eine
"leere Eingabe".

```
*****
*      EXPONENTIELLE GLAETTING ZWEITER ORDNUNG      *
*VON PER. 1 BIS PER. 11 ALFA = .60
*M.A.A3W= 2.64
*
*PERIODE BEOBACHT.GRUNDWT.TRENDWT.PROGNOSE ABS.ABW.*
* 15 233.20 *
* 16 235.20 *
* 17 238.30 *
* 18 241.10 *
* 19 244.80 *
* 20 243.30 *
* 21 251.80 *
* 22 251.73 *
* 23 253.53 *
* 24 243.30 *
*
*
*
*
*****
```

Bild 45:

```
*****
**ALFA = 0.3 **EG2
*
*WOLLEN SIE NUN
*
* 1 DAS TRAINING BEENDEN,
* 2 EIN NEUES BEISPIEL BEARBEITEN ODER
* 3 EINEN PARAMETER VERAENDERN
* (A, B, ALFA, TANF, TEND)?
*
*
*
*
*
*
*
*
*
*
*
*****
```

Bild 46:

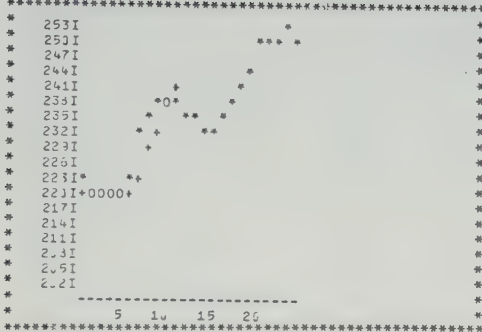


Bild 47:

```

*****
*
*      EXPONENTIELLE GLAETTUNG ZWEITER ORDNUNG
*
* VON PER. 1 BIS PER. 11 ALFA = .30
* M.A.ABW= 3.11
*
* PERIODE  BEOBACHT. GRUNDWT. TRENDWT. PROGNOSE ABS. ABW.
*
* 1 222.30 220.00 .01 221.01 2.29
* 2 220.70 221.17 .21 221.39 .69
* 3 219.10 221.04 .15 221.19 2.09
* 4 219.40 220.12 -.03 220.09 .69
* 5 219.40 219.73 -.09 219.64 .24
* 6 224.00 219.51 -.11 219.39 4.61
* 7 231.30 221.74 .29 222.14 9.75
* 8 235.70 227.01 1.17 228.19 7.50
* 9 237.20 232.02 1.34 233.37 3.32
* 10 238.50 239.66 2.14 237.71 .78
* 11 238.10 239.11 2.21 240.33 2.23
* 12 236.90 239.19 2.01
* 13 234.70
* 14 234.00
*****

```

```

*****
*
* **ALFA = 2.0 *TEND = 15 *EG2
*
*      EXPONENTIELLE GLAETTUNG ZWEITER ORDNUNG
*
* VON PER. 1 BIS PER. 11 ALFA = .30
* M.A.ABW= 3.11
*
* PERIODE  BEOBACHT. GRUNDWT. TRENDWT. PROGNOSE ABS. ABW.
*
* 15 233.20
* 16 235.20
* 17 236.30
* 18 241.00
* 19 244.80
* 20 249.90
* 21 251.90
* 22 251.70
* 23 253.90
* 24 249.30
*
*
*
*****

```

Der Student hat falsch eingegeben. ALFA muß zwischen 1 und 0 liegen.

Bild 48:

```

*****
**ALFA = 0.3 *EG2**
**
**SIE HADEN EIN ALFA EINGEGEBEN, DAS GROESSER ALS
**1 IST. DAS IST NICHT ERLAUBT. BITTE GEBEN SIE NEU
**EIN.
**
**
**
**
**
**
**
**
**
**
**
**
*****

```

Bild 49:

```

*****
**TEND = 19 *EG2**
**
**253I**
**251I***
**247I**
**244I**
**241I**
**238I**
**235I**
**232I**
**229I**
**226I**
**223I**
**220I+0000+
**217I**
**214I**
**211I**
**208I**
**205I**
**202I-----
**      5  10  15  20
*****

```

Bild 50:

```

*****
**TEND = 25 *EG2**
**
**253I**
**250I***
**247I**
**244I**
**241I**
**238I**
**235I**
**232I**
**229I**
**226I**
**223I**
**220I+0000+
**217I**
**214I**
**211I**
**208I**
**205I**
**202I-----
**      5  10  15  20
*****

```

Bild 51:

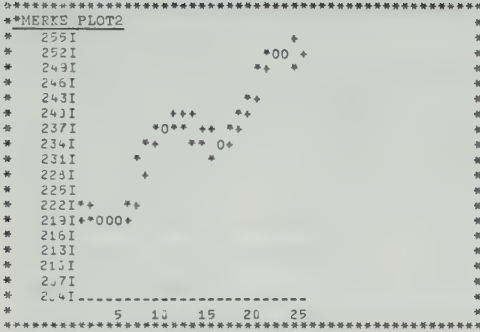


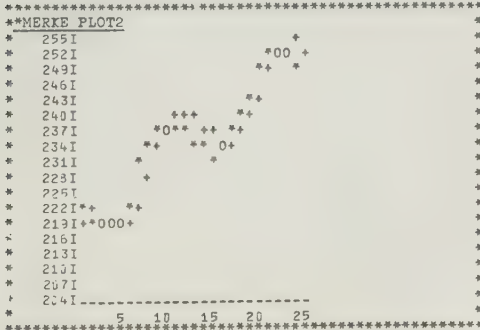
Bild 52:

```

*****
**LOESCHE GRAPHIK
**
**SIE HABEN BEREITS DREI BILDER DURCH *MERKE GE-
**SPEICHERT:
**= PLOT2
**= GRAPHIK
**= PLOT2
**WENN SIE NUN EIN ANDERES BILD SPEICHERN WOLLEN,
**SO MUESSEN SIE EINES DER FRUEHEREN BILDER LOESCHEN:
**(*LOESCHE....).
**
**
**
**
**
**
**
**
**
**
**
*****

```

Bild 53:



Der Student erhält nach dem Hinweis auf eine fehlerhafte Eingabe wie bei einem *RETURN wieder jenes Bild, bei dem die Eingabe gemacht wurde.

Bild 54:

```

*****
*      EXPONENTIELLE GLAETTUNG ZWEITER ORDNUNG*
*VON PER. 1 BIS PER. 25      ALFA = .33
*M.A.ABW= 3.36
*
*PERIODE  BEOBACHT.GRUNDWT.TRENDWT.PROGNOSE ABS.ABW.*
* 1 222.30 220.30 .01 222.01 2.23
* 2 224.70 221.17 .21 221.39 .69
* 3 219.10 221.04 .15 221.19 2.09
* 4 219.40 220.12 -.03 220.09 .69
* 5 219.40 219.73 -.09 219.64 .24
* 6 224.00 219.51 -.11 219.39 4.60
* 7 231.00 221.74 .29 222.04 9.75
* 8 235.70 227.01 1.17 229.19 7.50
* 9 237.20 232.02 1.84 233.87 3.32
* 10 238.50 235.56 2.14 237.71 .73
* 11 238.10 238.11 2.21 240.33 2.23
* 12 236.90 239.19 2.01 241.21 4.31
* 13 234.00 239.11 1.62 240.64 5.94
* 14 234.00 237.61 1.09 238.70 4.70
* 15 233.20 236.30 .67 236.37 3.77
*****
**TEND = 30 **EG2
*
*      EXPONENTIELLE GLAETTUNG ZWEITER ORDNUNG*
*VON PER. 1 BIS PER. 25      ALFA = .33
*M.A.ABW= 3.36
*
*PERIODE  BEOBACHT.GRUNDWT.TRENDWT.PROGNOSE ABS.ABW.*
* 16 235.20 235.05 .33 235.38 .18
* 17 238.30 235.28 .31 235.60 2.69
* 18 241.00 236.37 .55 237.53 3.46
* 19 244.00 239.30 .56 240.17 4.62
* 20 249.90 242.53 1.28 243.81 6.08
* 21 251.80 246.91 1.83 248.75 3.04
* 22 251.70 250.30 2.10 252.41 .71
* 23 253.50 252.04 2.04 254.09 .53
* 24 249.30 253.79 1.98 255.78 6.48
* 25 252.47 252.47 1.40 253.88
*
*
*
*****

```

Durch die Eingabe eines größeren TEND als die Anzahl der vorhandenen Beobachtungswerte erstellt der Student eine Prognose.

Bild 55:

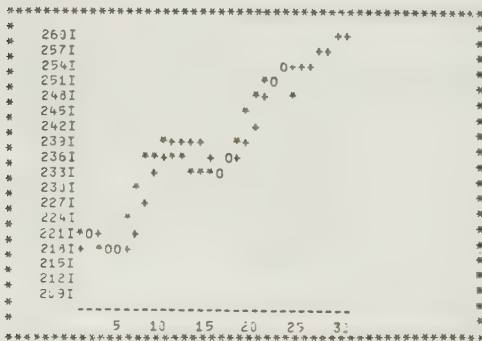


Bild 57:

```
*****
**ZEIGE PLOTG1**
**
**SIE HABEN FOLGENDE 24 VERGANGENHEITSWERTE (MONATS-
**UMSAETZE DES ARTIKELS "A 412" UND SOLLEN NUN DEN
**UMSATZ FUER DIE NAECHSTEN DREI MONATE PROGNOSTI-
**ZIEREN.
**
**MONAT    JAHR    UMSATZ    MONAT    JAHR    UMSATZ
**OKTOBER  1970    222.3    OKTOBER  1971    234.7
**NOVEMBER 1970    220.7    NOVEMBER 1971    234.0
**DEZEMBER 1970    219.1    DEZEMBER 1971    233.2
**JANUAR   1971    219.4    JANUAR   1972    235.2
**FEBRUAR  1971    219.4    FEBRUAR  1972    238.3
**MAERZ    1971    224.0    MAERZ    1972    241.0
**APRIL    1971    231.8    APRIL    1972    244.8
**MAI      1971    235.7    MAI      1972    249.9
**JUNI     1971    237.2    JUNI     1972    251.8
**JULI     1971    238.5    JULI     1972    251.7
**AUGUST   1971    238.1    AUGUST   1972    253.5
**SEPTEMBER 1971    236.9    SEPTEMBER 1972    249.3
*****
```

Bild 58:

```
*****
**ZEIGE PLOT2**
**
** 253I
** 250I      **+O+
** 247I      *+
** 244I      *+
** 241I      *+
** 238I      *+
** 235I      *+
** 232I      *+
** 229I      *+
** 226I      *+
** 223I*+
** 220I*+000++
** 217I
** 214I
** 211I
** 208I
** 205I
** 202I-----
**      5  10  15  20  25
*****
```

Bild 59:

```
*****
**ENDE
** 255I
** 252I      *00+
** 249I      *+
** 246I      *+
** 243I      *+
** 240I      *+
** 237I      *+
** 234I      *+
** 231I      *+
** 228I      *+
** 225I      *+
** 222I*+
** 219I*+000+
** 216I
** 213I
** 210I
** 207I
** 204I-----
**      5  10  15  20  25
*****
```

Mit *ENDE verläßt der Student die Trainingslektion.

2. Der Begriff "Computerunterstütztes Entscheidungstraining"

- Entscheidungstraining

Ziel des CET ist es, den Studenten in Entscheidungssituationen zu versetzen und ihn den Entscheidungsprozeß üben zu lassen. Dabei soll der Student den Einsatz verschiedener Verfahren bei der Entscheidungsvorbereitung kennenlernen.

- Computerunterstützt

Dieses Training ist wegen des hohen Aufwandes für die Berechnung der Modelle manuell nicht durchführbar. Im CET werden daher die Routinetätigkeiten, die für den Lernvorgang ohne Bedeutung sind, vom Rechner übernommen.

3. Charakteristika des CET

Im Folgenden sollen die wichtigsten Merkmale der Methode CET erläutert werden:

3.1. Simulation der Praxis

Im CET sollen wirkliche Systeme (Unternehmung oder Teilbereiche der Unternehmung) in der Weise simuliert werden, daß der Student lernt, formale Verfahren auf praxisnahe Entscheidungssituationen anzuwenden. Diese Lehrmethode kommt somit dem Wesen des Praktikums in den Naturwissenschaften sehr nahe, indem der Student lernt, das zur Verfügung stehende Instrumentarium so anzuwenden, daß er auch praktische Probleme lösen kann. Es wurde mit dem CET ein "Quasi-Labor" geschaffen, in dem der Student sein wissenschaftliches Instrumentarium anwenden kann.

Diese Art des Computereinsatzes im Unterricht entspricht auch der Aussage von Myers:

"The computer provides teaching capabilities and potential which are largely untapped in the corporate environment. The use of the computer for management, technical and professional training at all levels should increase exponentially in this decade"¹⁾.

3.2. Nutzung des Computers

Das CET umgeht die Schwierigkeiten, die bei der Übung komplexer Verfahren wegen des hohen Arbeitsaufwandes entstehen, durch die Nutzung der besonderen Fähigkeiten elektronischer Rechner:

- hohe Rechenkapazität

Im CET nimmt der Rechner dem Studenten den hohen Rechenaufwand bei der Lösung betriebswirtschaftlicher Probleme ab. Durch die Ausnutzung der spezifischen Fähigkeiten des Computers, der hohen Rechenkapazität, wird es dem Studenten ermöglicht, sich auf das Wesentliche zu konzentrieren.

- hohe Speicherfähigkeit

Betriebswirtschaftliche Problemstellungen sind aber auch durch eine große Datenmenge gekennzeichnet. Die Datengewinnung, -organisation und -suche ist oft eine der größten Schwierigkeiten. Hier bietet sich der Rechner wegen seiner großen Speicherfähigkeit an, dem Studenten den Umgang mit großen Datenmassen zu erleichtern und ihn darin zu schulen.

¹⁾ Myers, E., Effective vs. Efficient Computing, Datamation 18 (1972) 6, S. 75.

Dies ist besonders unter Berücksichtigung der Zunahme der Bedeutung von Datenbanken in der Praxis wichtig.

- Dialogfähigkeit

Im CET werden betriebswirtschaftliche Probleme in einem MMK-Prozeß gelöst. Dies kommt der von Mertens beschriebenen Tendenz der Praxis entgegen, Entscheidungshilfen on-line einzusetzen¹⁾, da der Student bereits auf der Universität lernt, MMK-Verfahren zu verwenden.

3.3. Lehrmethoden-"Mix"

Wie wir bereits in Punkt 1 festgestellt haben, sind verschiedene Methoden zur Lehre bestimmter Lehrstoffe unterschiedlich geeignet²⁾. Das CET wurde zur Lehre der Verfahrensanwendung in praxisnahen Fällen entwickelt. Es hat sich aber insbesondere beim RUU herausgestellt, daß die Benutzung einer einzigen Lehrmethode wenig zielführend ist³⁾.

Es ist im CET daher vorgesehen, dem Studenten folgende Lehrstrategien anzubieten:⁴⁾

-
- 1) Vgl. Mertens, P., Der Einfluß der elektronischen Datenverarbeitung auf Entscheidungsfindung und Entscheidungsprozeß, in: Jacob, H. (Hrsg.), Zielprogramm und Entscheidungsprozeß in der Unternehmung, Schriften zur Unternehmensführung, Band 11, Wiesbaden 1970, S. 83 f.
 - 2) Vgl. dazu auch Miller, N., The Computers Role in the Educational Process, MIT, MS-Thesis Cambridge 1971, S. 15 ff.
 - 3) Vgl. dazu auch Rockart, J.F., An Integrated Use of Available Resources (Student, Professor and Technology) in the Learning Process, MIT, Working Paper 601-72 Cambridge 1972.
 - 4) Vgl. Punkt 4.

- Literaturstudium,
- Lehrveranstaltungen,
- Rechnerunterstützte Selbsttests,
- Rechnerunterstützte Informationssuche,
- Training und
- Lernersteuerung.

Der Entschluß zur Verwendung eines Methoden-"Mixes" wurde durch folgende Faktoren beeinflußt:

- Immer wieder zeigt sich bei der Erprobung neuer Lehrverfahren, daß isolierte Lehrmethoden weniger erfolgreich sind als Verfahren, bei denen mehrere Teilstrategien kombiniert werden. Auf dieses Phänomen weist vor allem Rockart hin¹⁾.
- Lehrmethoden, die mehrere Teilstrategien verbinden, wurden von den Studenten in einer Befragung eindeutig isolierten Lehrstrategien vorgezogen²⁾.
- Die optimale Lehrmethode für eine bestimmte Lernsituation ist nicht für alle Schüler anhand eines allgemeinen Modells bestimmbar.

3.4. Freiheit des Lernenden

Das CET ist durch eine große Bewegungsfreiheit des Studenten innerhalb des Kurses gekennzeichnet. Dies entspricht einerseits der Philosophie des CET, die Praxis möglichst gut abzubilden und damit dem Studenten auch die Freiheitsgrade der Praxis zu geben, andererseits scheint es uns auch unter didaktischen Gründen sinnvoll zu sein, da es derzeit noch wenig Anhaltspunkte zur Bestimmung der optimalen Lernstrategie für einen Studenten gibt und ein Student auf der Universität aufgrund seiner langen Lernerfahrung die für ihn geeignete Strategie am ehesten noch selbst bestimmen kann. Diese Freiheit des Studenten äußert sich in folgenden Punkten:

1) Rockart, J.F., An Integrated Use, a.a.O., S. 3 ff.

2) Vgl. dazu Punkt 5,

- Freiheit bei der Wahl des Lernweges

Im Programmierten Unterricht wird versucht, dem Studenten in Abhängigkeit von seinen Antworten einen Lernweg vorzugeben. Da nur eine sehr begrenzte Zahl von Lernwegen mit vernünftigem Programmieraufwand realisierbar ist, müssen die Studenten klassifiziert werden. Es gibt aber sehr wenig gesichertes Wissen darüber, wie solche Klassen zu bilden sind und wie der Lernstoff darin angeboten werden soll. Beim derzeitigen Stand der Forschung auf diesem Gebiet dürfte es am vorteilhaftesten sein, die Entscheidung dem Studenten zu überlassen, welche Informationen er in bestimmten Situationen braucht, um einen möglichst großen Lernfortschritt zu machen.

- Freiheit bezüglich der Lerngeschwindigkeit

Gegenüber den Massenveranstaltungen bringt das CET den Vorteil eines weitgehend individuellen Lerntempos. Einschränkungen ergeben sich allerdings dann, wenn für bestimmte Kursteile Gruppenunterricht eingesetzt wird.

Eine weitere Beschränkung der Freiheit ergibt sich durch die Begrenzung der Rechenzeit, die ein einzelner Student zur Verfügung hat. Prinzipiell ist die Lerngeschwindigkeit aber dem einzelnen Studenten überlassen.

- Freiheit bei der Wahl der Lehrmethode

Da für verschiedene Studenten verschiedene Lehrmethoden unterschiedlich geeignet sind¹⁾ und ein formales Modell für die Zuweisung einzelner Studenten zu bestimmten Methoden nicht existiert, ist es für eine hohe Lernwirksamkeit notwendig, die Entscheidung über die Lehrmethoden dem Lernenden zu überlassen. Dieses Prinzip wurde bei der Konzeption der Lernersteuerung stets beachtet.

1) Unsere Untersuchungen weisen darauf hin, daß Studenten unterschiedliche Lehrmethoden bevorzugen.
Vgl. dazu auch Dixon, G.T., a.a.O., S. 34 ff.

- Freiheit bei der Wahl des Lernstoffes

In einem gewissen Maße kann der Student auch den Lernstoff selbst auswählen. Hat er die Kursziele erreicht, so kann er die noch verbleibende Rechenzeit zur Vertiefung des erworbenen Wissens benutzen. Er hat nach Erreichung der Kursziele völlige Freiheit, sich selbst Schwerpunkte zu setzen.

3.5. Lerndatenerhebung

Zur Austestung von CET-Kursen muß eine automatische Aufzeichnung der Lerndaten (wie Antwortzeit, Studenteneigabe, Lernweg) durchgeführt werden, die dem Kursautor einen leichten Überblick über Erfolg und Schwierigkeiten bei der Durcharbeitung des Kurses geben kann.

4. Lehrstrategien des Computerunterstützten Entscheidungsstrainings

Die Kombination mehrer Lehrstrategien¹⁾ ist ein wesentlicher Bestandteil des CET. Die einzelnen Strategien und ihre Verknüpfungen werden daher in diesem Punkt eigens behandelt.

Bereits die ursprüngliche Konzeption des CET sah einen Lehrmethoden-"Mix" vor. So entstand ein CET-Versuchssystem, in dem der Student mit Hilfe eines RPU auf das Training vorbereitet wurde. Im Rahmen dieses RPU stand dem Lernenden zusätzlich ein Lexikon zur Verfügung, wo er Begriffe "nachschlagen" konnte.

1) Wir verwenden hier das etwas hochtrabende Wort "Lehrstrategie", da es nach der Terminologie in der RUU-Literatur die einzelnen Teile des CET am besten charakterisiert.

Aufgrund der Erfahrungen mit dem Versuchssystem wurde das Strategienspektrum auf die in Abb. 1 dargestellten Lehrmethoden erweitert. Teile daraus sind bereits verwirklicht worden, andere Teile bestehen im Konzept.

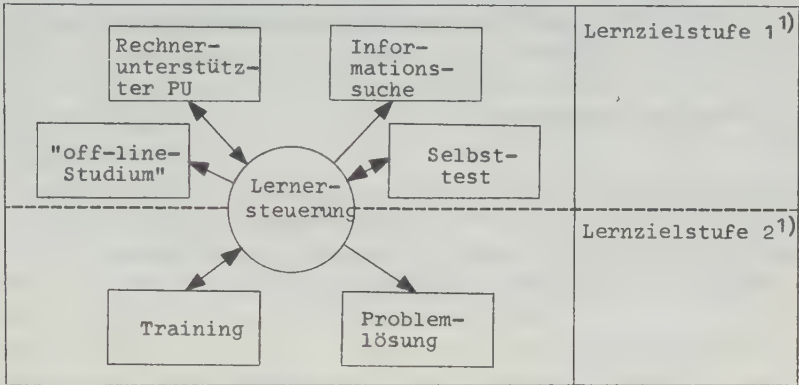


Abb. 1: Das Zusammenwirken der CET-Strategien.

Die Lernersteuerung stellt in Abb. 1 den zentralen Punkt dar, auf den sich sowohl Lehrmethoden der Stufe 1 als auch der Stufe 2 beziehen. Sie kann den Studenten auf alle angegebenen Lehrstrategien hinweisen, erhält aber nur von jenen eine Rückmeldung, mit denen sie durch eine zweiseitig gerichtete Kante verbunden ist.

Es wird versucht, bei der Erstellung der Strategien, vom reinen "frame-by-frame-writing" abzugehen und so eine möglichst kostengünstige Programmerstellung zu erreichen. Programmteile, die sich in einer Strategie immer wiederholen, sollen zu Ablaufroutrinen zusammengeschlossen werden. Im günstigsten Fall (z.B. beim Lexikon) kann dadurch eine Kurseingabe per "data format" erreicht werden. Ist eine so starke Vereinfachung nicht möglich, sollen wenigstens

1) Vgl. Teil 1.

oft benötigte Codemengen zu Makrobefehlen zusammengefaßt werden.

Im Folgenden werden die einzelnen Strategien in ihrer Funktion beschrieben. Dabei wird auf ihre derzeitige Realisierung und auf Realisierungskonzepte hingewiesen.

4.1. "Off-line-Studium"

Unter dem Begriff "Off-line-Studium" seien jene Lehrmethoden zusammengefaßt, die nicht vom Rechner unterstützt werden. Der Student soll mit ihrer Hilfe betriebswirtschaftliche Verfahren und Modelle theoretisch kennenlernen.

4.1.1. Literaturstudium

Arbeitsblätter, Skripten, Lehrbücher und Programmierter Unterricht in Buchform stehen dem Lernenden zur Aneignung des Grundwissens zur Verfügung. Hinweise darüber, welche Stellen er daraus für den Kurs benötigt, erhält er von der Lernersteuerung.

4.1.2. Lehrveranstaltungen

Lehrveranstaltungen eignen sich sehr gut zu einer Verbindung mit dem Training. Sie haben dabei weniger den Zweck, Grundlagenwissen zu vermitteln, als vielmehr, einen Diskussionsrahmen für die Erfahrungen der Studenten beim Training zu bieten. Für diese Zielsetzung dürfte das Seminar am besten geeignet sein.

4.2. Informationssuche

4.2.1. Aufgaben

Der Student muß während des Kurses jederzeit - außer während des Selbsttests - Gelegenheit haben, kleine Wissenslücken durch "Nachschlagen" in einem Lexikon beseitigen zu können. Er soll dazu entweder mit dem Suchbegriff direkt auf das Lexikon zugreifen oder sich ein Inhaltsverzeichnis ausgeben lassen können.

Die Verarbeitung von Suchoperatoren wie beispielsweise "WOZU", "WANN" und "WIE" ist nach den Erfahrungen mit dem System ALPS am MIT¹⁾ aus Kostengründen nicht wünschenswert. Diese Operatoren können durch Hinweise ersetzt werden, wie sie in Lexika in Buchform üblich sind. Dies ist zwar nicht so komfortabel, dafür aber wesentlich kostengünstiger.

4.2.2. Realisierung

Das Lexikon besitzt eine sehr einfache Ablaufstruktur, die ohne Schwierigkeiten in einer Routine bereitgestellt werden kann. Die Eingabe des Autors kann auf

- die Angabe des Suchbegriffes (eventuell auch von Alternativen) und
 - die Erklärung
- beschränkt werden.

Im Rahmen des RPU-Teiles im CET-Versuchssystem steht dem Lernenden ein Lexikon zur Verfügung, das vom PUMA²⁾ bereit-

1) Vgl. Rockart, J., o.T., a.a.O., S. 33.

2) PUMA ist eine RUU-Sprache, die zur Erstellung des RPU-Teiles des CET herangezogen wurde. Eine Beschreibung von PUMA befindet sich in: Endres, C., PUMA - Programmierter Unterricht unter CD 3300 MASTER, Mitteilungsblatt des Rechenzentrums der Universität Nürnberg Nr. 13, Erlangen 1973.

gestellt wird. Diese Strategie besaß aber bisher eine eher geringe Bedeutung, da sie der Student nur im Rahmen des Programmierten Unterrichts benutzen konnte.

4.3. Rechnerunterstützter Programmierter Unterricht

4.3.1. Aufgaben

Die Erfahrungen mit dem CET¹⁾ haben gezeigt, daß der Einsatz von RPU nur in bestimmten Fällen sinnvoll ist. Es ist keineswegs anzustreben, die gesamte Ausbildung auf der Lernzielstufe 1 mit RPU zu bewerkstelligen.

Gegen die Verwendung des RPU auf breiter Basis sprechen folgende Argumente:

- Die Erstellung von Lernprogrammen "frame-by-frame" ist sehr zeitaufwendig.
- Lernmodelle, die mit einer großen Wahrscheinlichkeit einen hohen Lernerfolg bringen, sind nicht vorhanden.
- Die Lernenden scheinen den RPU anderen Lehrmethoden wie dem Seminar nicht vorzuziehen¹⁾.
- Die variablen Computerkosten für diese Lehrmethode sind hoch.

Der RPU ist dann einsetzbar, wenn das Lexikon zur Beseitigung von Wissenslücken nicht ausreicht und der benötigte Lehrstoff nicht so umfangreich ist, daß es sich lohnen würde, für das Nachschlagen in der Literatur eine Terminal-sitzung abzubrechen.

4.3.2. Realisierung

Mit dem zu fordernden System soll jede Art des Programmier-ten Unterrichts verwirklicht werden können. Zur Vereinfachung

1) Vgl. dazu Punkt 5.

der Programmierung kann der Autor für bestimmte immer wiederkehrende Befehlsfolgen Makros erstellen.

Sollen nur kleine Hilfeprogramme erstellt werden, ist es empfehlenswert, auf eine "high sophisticated" Programmierung zu verzichten und eine Darbietungsroutine zu schaffen, die beispielsweise wie in Abb. 2 ablaufen könnte.

Der Autor muß nur die Texte und Antworten in das System einspeichern und sie über die Routine abarbeiten lassen.

Im RPU setzt die Erreichung eines bestimmten Punktes im Programm meist die Erfüllung bestimmter Lernziele voraus. Diese können über den Lernzielfile an die Lernersteuerung gemeldet werden.

Der RPU wurde im Projekt CET ursprünglich allein zur Vermittlung des Grundlagenwissens eingesetzt. Es sind Lernprogramme für die Einführung in die Prognoserechnung und für einzelne Prognoseverfahren in der Sprache PUMA vorhanden. Die Zielsetzung des PU-Einsatzes hat sich aber inzwischen durch die Erkenntnisse aus den Versuchen in die bereits beschriebene Richtung verändert.

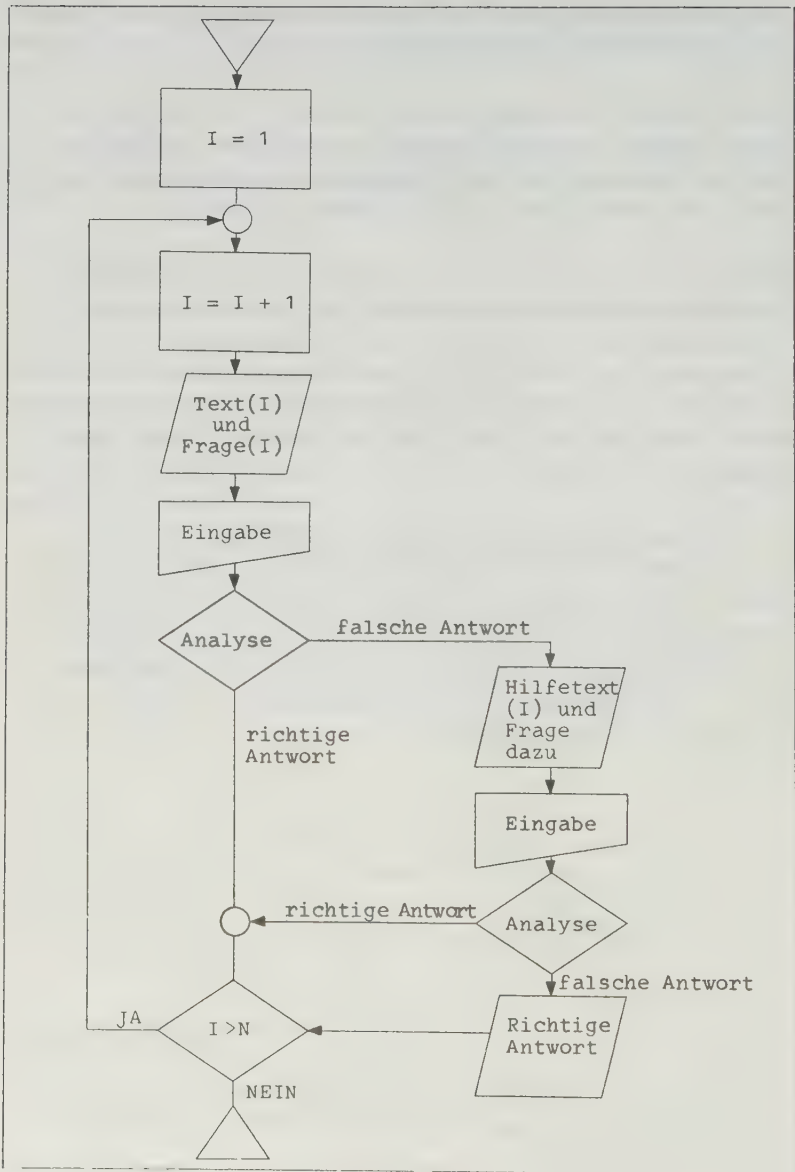


Abb. 2: Stoffdarbietungsroutine

Erklärung der Symbole¹⁾:



Programminterne Funktion (z.B.: Rechnen)



Ausgabe auf Datenendgerät



Entscheidungspunkt



Studenteneingabe



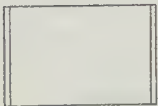
Konnektor



Programmanfang und -ende



Ablauflinie



Unterprogramm

1) Diese Symbolik gilt für die ganze Arbeit.

4.4. Rechnerunterstützte Selbsttests

4.4.1. Aufgaben

Selbsttests haben die Funktion, das Wissen des Adressaten auf der Lernzielstufe 1 zu ermitteln.

- Dies geschieht, bevor der Student zum ersten Mal mit dem Inhalt der entsprechenden Lektion in Berührung kommt. Die Testergebnisse bilden neben anderen Daten die Grundlage für die Lernersteuerung.
- Bevor der Student den Lernstoff der Stufe 2 zu bearbeiten beginnt, sollte er sich wieder einem Selbsttest unterziehen. Er kann daraus entnehmen, ob es für ihn bereits sinnvoll ist, das Gelernte anzuwenden, oder ob er noch zu große Wissenslücken aufweist. Der Test ist an dieser Stelle wesentlich für die Effizienz beim Training der Modelle.
- Der Student kann aber auch zu jedem anderen Punkt seines Lernprozesses eine individuelle "Standortbestimmung" vornehmen, um daraus die für ihn günstigsten Lernstrategien zu ermitteln.

Selbsttests sind vor allem auf der Lernzielstufe 1 wichtig, da der Student hier viele Möglichkeiten besitzt, sich Wissen anzueignen, und daher nur mit Hilfe eines Tests sein Wissensstand erhoben werden kann.

Auf der Stufe 2 ist es ungleich schwieriger, einen Wissenstest durchzuführen, da es sich hier um Lernziele handelt, die nicht durch Abfragen von Faktenwissen überprüft werden können. Es besteht für das CET aber auch keine unmittelbare Notwendigkeit, hier die Lernzielerreichung zu messen, da alle Studenten mit derselben Lehrmethode dieselben Möglich-

keiten besitzen und das Anwendungswissen nicht wiederum Voraussetzung für eine nächst höhere Ausbildungsstufe ist.

4.4.2. Realisierung

In RPU-Systemen können Selbsttests nur durch Programmierung jeder einzelnen Frage erstellt werden. Selbsttests sind zwar sehr einfach strukturierte Programme, erfordern aber bei dieser Art der Erstellung dennoch einen relativ großen Aufwand, da in jedem frame eine Reihe von gleichbleibenden Anweisungen immer wieder geschrieben werden muß.

Das Abspeichern der Testergebnisse auf einem Plattenfile, auf den auch die Lernersteuerung Zugriff haben muß, ist in den meisten Unterrichtssystemen nicht möglich. Doch auch in Systemen, die einen Plattenzugriff erlauben, müßte dieser in jedem Test neu programmiert werden.

Der Ablaufteil eines Selbsttests ist linear und kann somit ohne Schwierigkeiten von einer Routine übernommen werden. Sollen auch verzweigte Tests durchgeführt werden, so kann auch dies in einer Routine vorgesehen werden. Der Autor kann sich somit auf folgende Eingabe beschränken:

- Fragen-Texte

Für jedes Lernziel, das der Autor in einem Test überprüfen will, muß er eine oder mehrere Fragen mit Angabe der dazugehörigen Lernzielnummer eingeben. Die Selbsttestroutine übergibt bei Abarbeitung durch den Studenten Erfolgs- oder Mißerfolgsmeldungen über einen schülerindividuellen Lernzielfile an die Lernersteuerung.

- Antwortkontrolldaten

Dazu ist aber eine Analyse der Studentendaten nötig. Diese variiert von Frage zu Frage und muß somit ebenfalls vom Autor angegeben werden. Dazu sollen die

in Teil 4, Punkt 2.3.4., geforderten Antwortverarbeitungs-
routinen zur Verfügung stehen.

Bei der Eingabe eines Tests durch den Autor wird in einem
Kursbeschreibungsfile eine entsprechende Vormerkung durch-
geführt, aufgrund der für jeden Studenten, der mit diesem
Kurs arbeitet, der benötigte Lernzielfile reserviert wird.

Die Selbsttests sind derzeit im Rahmen des Programmierten
Unterrichts in PUMA verwirklicht. Sie dienen einerseits als
Test des Vorwissens, um dem Studenten einen Weg durch das
Lernprogramm vorschlagen zu können, und andererseits zur
Überprüfung, ob der Lernende schon genügend Wissen besitzt,
um mit dem Training zu beginnen. Eine Bereitstellung der
Testergebnisse auf dem Lernzielfile und eine Auswertung
durch die Lernersteuerung ist noch nicht verwirklicht.

4.5. Training

4.5.1. Aufgaben

Das Training bildet den Kern des CET. Es soll dem Studen-
ten die Möglichkeit bieten, Anwendungswissen zu erwerben.
Im Punkt 3 ist das Wesen dieser Methode bereits dargestellt
worden¹⁾.

4.5.2. Realisierung

Das Training muß meist Dialogschritt für Dialogschritt pro-
grammiert werden, da es nur wenige Möglichkeiten zu einer
Standardisierung und somit zur Erstellung von Ablauf routi-
nen bietet²⁾.

1) Vgl. dazu auch Reitbauer, F., a.a.O., Punkt 1 und 2.

2) Eine umfassende Beschreibung des Trainingsteils findet
sich in: Reitbauer, F., a.a.O., Punkt 2 und 4.

4.6. Problemlösung

4.6.1. Aufgaben

Die Problemlösungsstrategie hat sich aus dem Training ergeben. Hat der Lernende Probleme im Rahmen einer wissenschaftlichen Arbeit, kann er die Rechenverfahren, die im Training im Dialog zur Verfügung gestellt werden, zur Lösung heranziehen. Er kann zum Beispiel in der Prognoserechnung eigene Vergangenheitsdaten in den Rechner eingeben und diese beliebig mit den vorhandenen Verfahren analysieren.

4.6.2. Realisierung

Der Aufwand zur Bereitstellung von derartigen Problemlösungshilfen ist relativ gering, wenn entsprechende Trainingsprogramme zur Verfügung stehen. Die bereits dialogisierten Rechenroutinen können in kurzer Zeit in eine Problemlösungsstruktur eingebaut werden, die dem Studenten mehr Freiheit als das Training erlaubt.

4.7. Lernersteuerung

4.7.1. Aufgaben

Die Lernersteuerung hat die Aufgabe, den Studenten bei der Wahl eines Lernweges auf Vorschlagsbasis zu unterstützen. Da es beim derzeitigen Stand der Erziehungswissenschaften nicht möglich ist, für einen speziellen Schüler in einer bestimmten Lernsituation den besten Lernweg aufgrund eines formalen Modells zu bestimmen, ist es sinnvoll, ihm zwar einen Überblick über mögliche Wege zu geben, ihn aber diese frei wählen zu lassen.

Auch bei der Auswahl der Lehrmethoden sind die Kosten der Wissensvermittlung zu beachten, da andernfalls "Ausbildungsverluste" eintreten. Dies sei an folgendem Beispiel (Tab. 1) demonstriert, dessen Zahlenwerte zwar willkürlich gewählt wurden, das aber die Problematik gut erkennen läßt.

Lehrmethode	Effizienzwert	Kosten
Lehrbuch	8	100
RPU	9	1000

Tab. 1

Sind für den Studenten die Kosten irrelevant, wird er sich für den RPU entscheiden, von dem er sich eine kürzere Lernzeit erhofft. Da es aber für die Ausbildungsinstitution ein Optimierungsproblem ist, für welche Lehrformen sie ihr Budget einsetzt, müssen bei der Auswahl der Lehrmethoden auch Kosten Gesichtspunkte eingeführt werden. Dies kann beim CET am einfachsten über eine Begrenzung der gesamten Rechenzeit pro Kurs und Student geschehen, da die Kosten für die Rechenzeit den weitaus größten Anteil an den variablen Kosten haben. Es bleibt dem Studenten freigestellt, wie er diese Rechenzeit für die einzelnen Kursstrategien verwenden will. Die Lernersteuerung soll ihm auf Wunsch die bereits verbrauchte und die noch offene Zeit ausgeben. Er soll aber auch Angaben über die Rechenintensität der einzelnen Strategien bekommen. In der hypothetischen Kurssitzung, die im Teil 2, Punkt 1, dargestellt ist, sind Beispiele für die Ausgabe der Lernersteuerung enthalten.

4.7.2. Realisierung

Dieser Teil des CET ist bisher aufgrund mangelnder Fähigkeiten der vorhandenen Software nicht realisiert worden. Ein Konzept für die relativ einfach zu verwirklichenden Routinen ist bereits erstellt worden. Danach ist der Zeitaufwand für den Autor, der die Steuerung für einen bestimmten Kurs ein-

setzen will, verhältnismäßig niedrig. Er muß nur Daten über die Reihenfolge des Lehrstoffes, die Lernziele und die dazugehörigen Texte eingeben.

5. Erfahrungen mit dem CET

Im Rahmen des Projektes CET wurde für die Validierung von Kursmaterial und zur Gewinnung von grundlegendem Wissen über die Gestaltung von CET-Kursen ein Versuchsplan zur Erhebung von Faktoren entwickelt, die den Lernerfolg beeinflussen können. Zur Analyse der Art und der Stärke dieser Einflüsse sollen Dialogprotokolle und Befragungen verwendet werden.

Es wurde eine Voruntersuchung angestellt, die allerdings einer Reihe von Restriktionen unterlag, aufgrund derer die Resultate nur als erste Hinweise gewertet werden können. Einerseits ließ es die sehr knappe Rechenzeit am Universitätsrechenzentrum nicht zu, den Stichprobenumfang für statistisch signifikante Aussagen hinreichend groß zu gestalten, andererseits war die Teilnahme der Studenten vollkommen freiwillig, so daß ein beachtlicher Teil der Versuchspersonen den umfangreichen Fragebogen nicht ausfüllte. Es lagen somit für die Auswertung nur zehn Fragebogen vor.

Die Voruntersuchung war daher nicht geeignet, statistisch exakte Aussagen über das CET zu ermöglichen. Sie sollte vielmehr Anhaltspunkte für die weitere Entwicklung des CET bringen und dazu beitragen, die Erhebungsinstrumente zu verbessern, die für eine umfangreiche Untersuchung im Zusammenhang mit einem Einsatz des CET in einer Lehrveranstaltung benötigt werden¹⁾. Die Population wird in diesem Kurs

1) Vgl. Teil 5.

eine wesentlich bessere Ausgangsbasis für Analysen bieten, da die Studenten das Entscheidungstraining im Rahmen einer Pflichtveranstaltung absolvieren werden. Verzerrungen, die durch eine freiwillige Teilnahme an Experimenten und eine unvollständige Beantwortung der Fragebogen entstehen, können nahezu vollständig ausgeschaltet werden.

Wir stellen hier die wichtigsten Ergebnisse der Voruntersuchung dar und weisen - wo es möglich ist - auf ähnliche Untersuchungen hin. So wird teilweise auf die Ergebnisse einer Befragung von acht Studenten hingewiesen, die am Rechenzentrum der Universität Erlangen-Nürnberg einen dort erstellten rechnerunterstützten ALGOL-Kurs absolviert und denselben Fragebogen wie die "CET-Studenten" ausgefüllt haben.

5.1. Lernerfolg

Die Lernziele des CET reichen bis zur Kategorie Anwenden, teilweise aber auch bis zur Stufe Beurteilen. Es ist schwierig, die Erreichung dieser Lernzielkategorien zu messen; es wird sogar unmöglich, wenn entsprechend umfangreiche Verfahren getestet werden sollen und dazu dem Studenten der Rechner nicht zur Verfügung steht. Selbst wenn die Berechnungen vom Computer übernommen werden können, bleibt dennoch die Schwierigkeit, festzustellen, wann Lernziele auf dieser Ebene erreicht sind. In der Untersuchung des CET wurde die Lernzielerreichung im Kurs unter besonderer Berücksichtigung von Anwendungsfragen in einem schriftlichen Test erhoben. Die Ergebnisse deuten auf einen beträchtlichen Lernerfolg hin, der schon mit der ersten Fassung eines CET-Kurses erreicht werden konnte. Vor allem bei den anwendungsbezogenen Fragen konnte ein guter Wissenszuwachs festgestellt werden.

5.2. Beurteilung des CET durch den Studenten

Wesentlich aufschlußreicher als die Erhebung des Wissenszuwachses, für die uns keine geeigneten Instrumente zur Verfügung stehen, ist die Beurteilung des Kurses durch die Lernenden, da man davon ausgehen kann, daß diese eine große Erfahrung mit dem Lernen aufweisen.

Bei den folgenden Ausführungen ist zu beachten, daß der PU im experimentellen Kurs einen bedeutend höheren Anteil hatte als das Training, das von allen Studenten dem PU vorgezogen wurde. Man kann daher von den Versuchsergebnissen nicht unmittelbar auf die derzeitige Version des CET schließen, in der das Training den weitaus größten Anteil am gesamten Kursumfang hat.

Die Erhebungen ergaben, daß die Studenten überwiegend dem Seminar den Vorzug geben, wenn ihnen folgende Alternativen zur Wahl gestellt werden:

- Lehrbuch,
- PU in Buchform,
- RUU,
- Vorlesung und
- Seminar/Übung.

Werden den Lernenden Kombinationen von jeweils zwei dieser Lehrverfahren vorgeschlagen, so ziehen die Studenten die Alternative Lehrveranstaltung gemeinsam mit RUU den anderen Möglichkeiten vor. Bei Dreierkombinationen entscheiden sich die Studenten für die Lehrveranstaltung gemeinsam mit PU in Buchform und RUU.

Diese Ergebnisse wurden auch durch die Untersuchungen im Zusammenhang mit dem ALGOL-Kurs im Wesentlichen bestätigt.

Auf die Frage, ob die Verwendung einer einzigen, von zwei oder von drei Methoden vorzuziehen sei, entschied sich der

Überwiegende Teil (7 von 10) für die Kombination von drei Lehrverfahren.

5.3. Analyse des CET

Beurteilung der Strategien

Vor allem aus den Antworten auf die Frage "Was hat Ihnen im Rahmen des Kurses besonders gut gefallen?" ließ sich eine äußerst positive Einstellung der Studenten gegenüber dem Training ableiten. Mehr als die Hälfte der Versuchspersonen erwähnte diese Strategie.

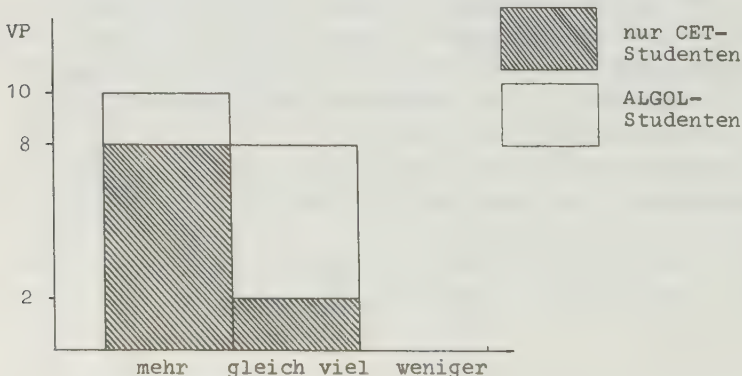
Individualität des Lernenden

Die Studenten fühlten sich durch den Kurs teilweise etwas eingeengt, doch dürfte sich diese Aussage hauptsächlich auf den PU-Teil bezogen haben, wie aus anderen Fragen hervorging¹⁾.

Frage:

Wieviel Freiheit sollten die Studenten in zukünftigen Kursen haben (im Vergleich mit jetzt)?

Beantwortung:



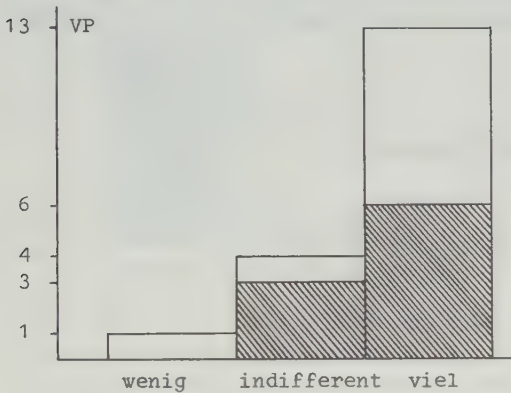
1) Ähnliche Ergebnisse für einen RPU-Kurs erhielt auch Dixon, G.T., a.a.O., S. 31.

Das individuelle Lerntempo beim RUU wurde überwiegend positiv gewertet.

Frage:

Brachte Ihnen das individuelle Lerntempo beim Computerunterstützten Unterricht Vorteile?

Beantwortung:

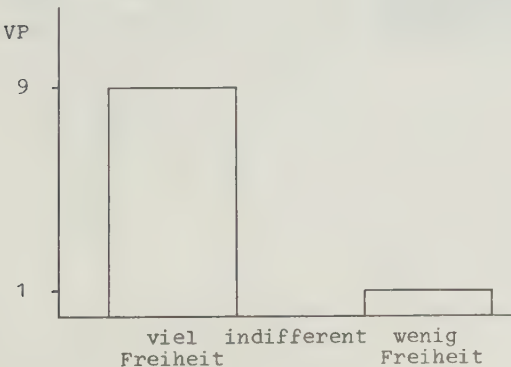


Die Freiheit im Training scheint den Lernenden meist zu genügen.

Frage:

Konnten Sie mit den Prognosemodell nach Ihrem Belieben spielen?

Beantwortung:



Lehrmedium

Die Einstellung der Studenten zum Rechner war meist positiv, wenn auch der verwendete Computer als störungsanfällig bezeichnet wurde.

Die Reaktionszeiten des Rechners wurden sowohl für den PU-Teil als auch für den Trainingsteil überwiegend gut beurteilt. Dagegen wurde die konzentrationsfeindliche Umgebung des Adressatenplatzes als Störung für den Lernerfolg empfunden.

Frage:

Beeinflußten die Geräusche im Terminalraum den
L e r n e r f o l g ?

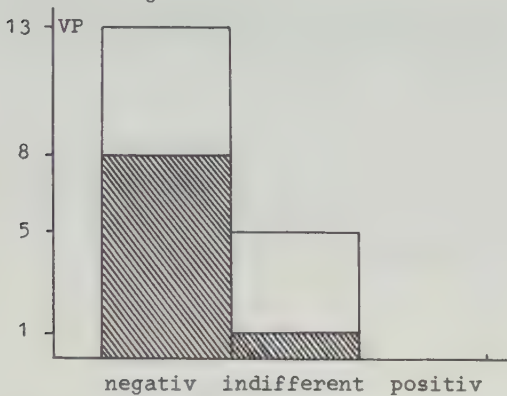
Beantwortung:



Frage:

Beeinflussten die Geräusche im Terminalraum Ihre
L e r n b e r e i t s c h a f t ?

Beantwortung:



Frage und Beantwortung (durchschnittlicher Rangwert):

Ordnen Sie bitte die folgenden Alternativen nach der Möglichkeit, sich auf den Lernstoff zu konzentrieren. Geben Sie dem Ort mit der höchsten Konzentrationsmöglichkeit eine 1 (usw.).

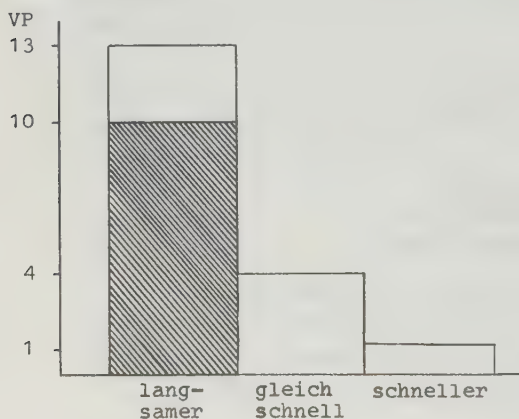
5,9	Straßenbahn
4,6	Terminal
1,3	Studierzimmer
3,6	Vorlesung
2,8	Lesesaal in der Bibliothek
2,8	Übung/Seminar

Das Ergebnis folgender Frage spricht stark gegen die Präsentation von großen Textmengen, zumindest auf den von uns verwendeten Bildschirmen.

Frage:

Konnten die Texte auf dem Bildschirm in normaler Leseschwindigkeit gelesen werden?

Beantwortung:



5.4. Zusammenfassung

Als Ergebnisse der Voruntersuchung, die zum Teil auch durch die Resultate ähnlicher Experimente bestätigt werden¹⁾, lassen sich folgende Hinweise angeben:

- Das CET wird von den Studenten als wertvolle Erweiterung der Lehrmethoden betrachtet.
- Die Trainingsstrategie wird als besonders positiv erwähnt.
- Die Kombination mehrerer Lehrmethoden, bei welcher der RUU enthalten ist, wird anderen Alternativen vorgezogen.
- Der Studentenplatz entspricht den Erfordernissen des Unterrichts nicht ausreichend.

1) Vgl. dazu vor allem Rockart, J.F., An Integrated Use, a.a.O., S.1 ff.

D r i t t e r T e i l

Realisierungsexperimente

=====

Im Rahmen des Projektes CET wurde eine Reihe von Experimenten durchgeführt, an denen wir die Eignung verschiedener Softwaresysteme zur Realisierung der CET-Strategien untersucht haben. Die Versuche bezogen sich auf

- PLANIT,
- PUMA,
- APL und das
- Trainingssystem (TS).

Sie bilden einen Teil der Untersuchungen, die wir zur Formulierung des Anforderungsprofiles an die Software (im Teil 4) angestellt haben, und werden dort nach funktionalen Kriterien interpretiert. Auf diese Weise können wir die Realisierungsmöglichkeiten den Anforderungen direkt gegenüberstellen. Wir beschränken uns in diesem Teil auf Erklärungen, die unmittelbar zum Verständnis der Beispiele erforderlich sind, und erläutern durch Verweise die Beziehungen der Experimente zur allgemeinen Behandlung der Realisierungsmöglichkeiten. Eine Beschreibung der Syntax der Sprachen oder eine detaillierte Programmbeschreibung sind in diesem Rahmen nicht möglich.

1. Softwaresysteme zur Erstellung von CET-Programmen

Zur Realisation von CET-Kursen stehen grundsätzlich drei Arten von Softwaresystemen zur Verfügung:

- spezielle Unterrichtssysteme sowie
- On-line- und
- Off-line-Programmiersysteme.

1.1. Unterrichtssysteme

Es existieren derzeit weit über hundert verschiedene Unterrichtssysteme¹⁾. Die Hardwareabhängigkeit der meisten dieser Systeme schränkt aber die Anzahl der für den RUU-Benutzer tatsächlich verfügbaren Sprachen stark ein. Für das Projekt CET standen mit ICU/PLANIT²⁾ ein sehr komfortables und mit PUMA³⁾ ein relativ flexibles und vor allem kostengünstiges RUU-System zur Auswahl. Diese beiden Systeme charakterisieren den derzeitigen Stand der Software für den computerunterstützten Unterricht sehr gut.

- PLANIT kann als eines der höchstentwickelten Systeme zur Realisierung von Programmierem Unterricht bezeichnet werden.
- PUMA dagegen ist ein allgemeineres - wenn auch dadurch für die Programmierung teilweise aufwendigeres - System, das aber vom PU abweichende Lehrmethoden leichter als reine PU-Systeme zu realisieren erlaubt.

In beiden Systemen wurden Versuche durchgeführt, die hier beschrieben werden.

1.2. On-line-Programmiersysteme

Als Alternativen zu reinen Unterrichtssystemen wurden Dialogprogrammiersprachen wie BASIC und APL in Erwägung gezogen. Diese Sprachen sind zwar nicht zur Erstellung von Dialogprogrammen, sondern zur Erstellung von Programmen im Dialog geschaffen worden, haben aber aufgrund ihrer interaktiven Verarbeitungsweise eine Anzahl von Makros zur Dialog-Ein-/Ausgabe, die in Off-line-Sprachen wie FORTRAN

1) Vgl. Zinn, K., a.a.O., S. 19 ff.

2) Butler, A.K., Frye, C.H., PLANIT Language Reference Manual, SDC-TM-(L)-4422/002/01 Santa Monica 1970.

3) Endres, C., a.a.O.

Die unter 2) und 3) genannten Unterlagen wurden auch für die weiteren Ausführungen zugrundegelegt.

icht zur Verfügung stehen.

m Schulungszentrum der IBM Deutschland in Stuttgart hatten
ir die Möglichkeit, ein kleines Trainingsprogramm in APL
u erstellen und einen Dialogablauf zu untersuchen.

.3. Off-line-Programmiersysteme

leben den Systemen, die speziell für Unterrichtsdialoge ge-
schaffen wurden, und interaktiven Programmiersystemen gibt
es auch die Möglichkeit, das Training in einer Off-line-
Programmiersprache wie FORTRAN zu schreiben.

Auf diese Sprache wurde zur Verwirklichung des Trainings
im CET zurückgegriffen. Zur Erleichterung der Programmie-
rung wurden Routinen für häufig wiederkehrende Tätigkeiten,
vor allem für die Dialog-Ein-/Ausgabe, erstellt. Teilweise
war es notwendig, mit COMPASS-Unterprogrammen¹⁾ verschiede-
ne Programmteile zu schaffen. Es entstand dadurch das TS²⁾,
in dem ein hier ebenfalls beschriebenes Trainingsprogramm
realisiert wurde.

2. Experimente

2.1. PLANIT

Mit einem Fernschreibterminal, das von der Siemens AG zur
Verfügung gestellt wurde, war das Projekt CET über Datex
an die Siemens 4004/45 des Projektes CUU in Freiburg an-
geschlossen. Neben einigen Experimenten mit PU-Programmen
wurde ein Trainingsprogramm erstellt, das im folgenden be-
sprochen wird.

1) COMPASS ist die Assemblersprache der im Rechenzentrum
der Universität Erlangen-Nürnberg installierten Rechenan-
lage Control Data 3300.

2) Vgl. dazu Reitbauer, F., a.a.O., Punkt 3.

2.1.1. Didaktogramm

- Dialogablauf

Siehe Abb. 3.

- Verfahrensbeschreibung

Siehe Punkt 2.4.1.

- Befehle

(1) Steuerung des Ablaufes

- < param..... Sprung zur Parameterbestimmung
- < daten..... Sprung zur Generierung oder Eingabe neuer
Daten
- < eg3..... Durchrechnung des Verfahrens
- < go(y)..... Sprung auf beliebigen Label (y)

(2) Dienstleistung

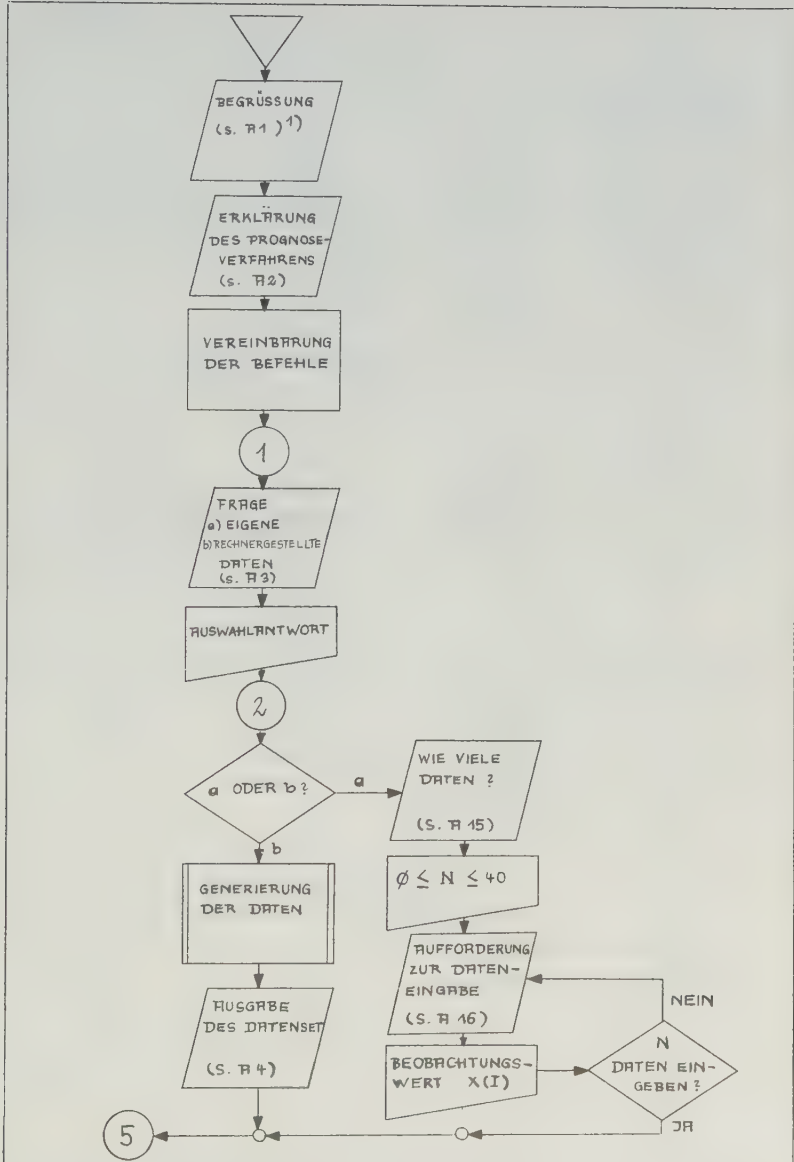
- < inhalt..... Inhaltsverzeichnis
- < befehle..... Befehlsliste

(3) Zuweisungsbefehle

- < al..... Glättungsparameter
- < a..... Grundwert
- < b..... Trendwert
- < c..... Trendwert der Trendwerte
- < tanf..... Beginnperiode der Glättung
- < tend..... Endperiode der Glättung

Diese Befehle stehen neben den in PLANIT generell vorhandenen Kommandos zur Verfügung (vgl. Teil 4, Punkt 2.3.5.2.1.).

Dialogablauf:



1) A i ist ein Ausgabeblock, der im Dialogbeispiel als solcher gekennzeichnet ist.

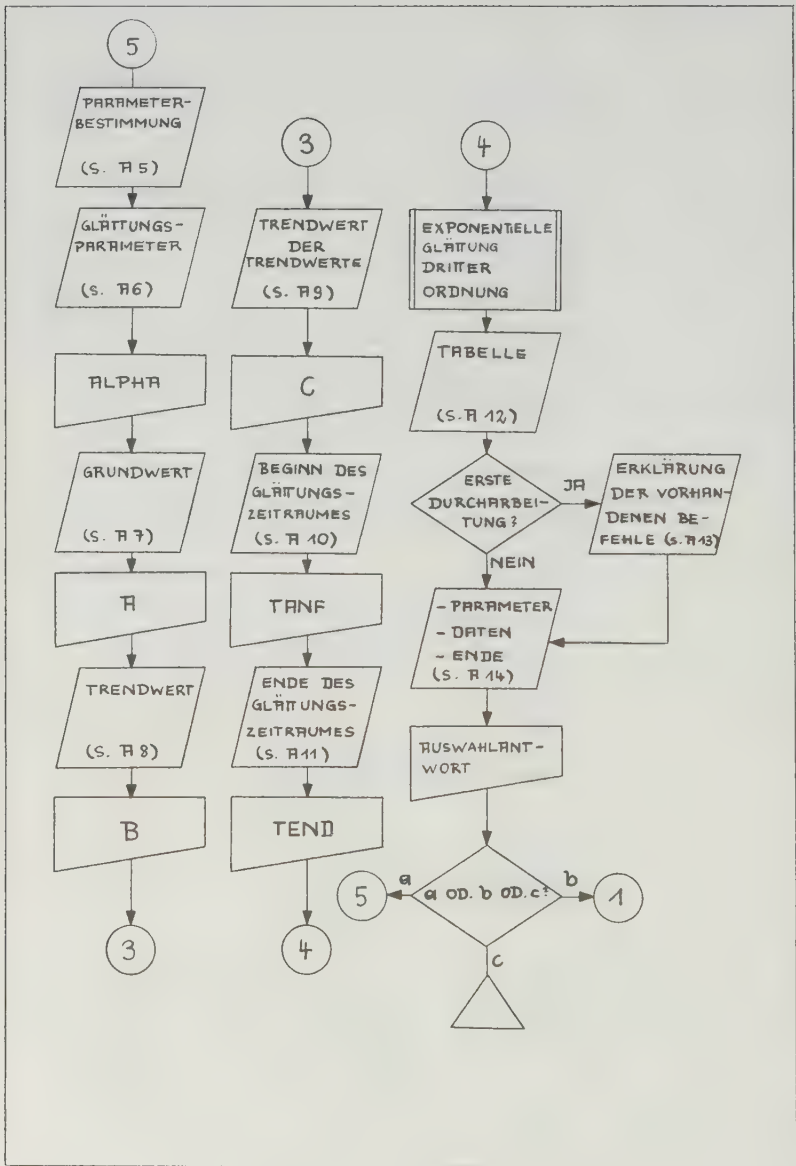


Abb. 3: Trainingsprogramm für exponentielle Glättung dritter Ordnung

2.1.2. Programm

Das hier dargestellte PLANIT-Programm ist wegen seiner relativ einfachen Struktur weitestgehend selbsterklärend¹⁾.

Wir beschränken uns daher auf kurze Hinweise zum Inhalt einzelner frames und Begründungen bestimmter Vorgehensweisen bei der Programmerstellung.

In den frames 1 und 1.01 wird der Lernende begrüßt (zur Textausgabe in PLANIT s. Teil 4, Punkt 2.3.3.2.1.). Er erhält sofort anschließend (in frame 1.1) eine Erklärung des Modells. Frame 1.3 dient primär der Definition der Felder und der Befehle.

Man sieht an diesem Beispiel sehr gut, wie einfach in PLANIT Studenten-Befehle definiert werden können. Allerdings handelt es sich im Beispiel um Kommandos, die keine sehr komplexe Verarbeitung erfordern.

Frame 1.5 bietet dem Studenten eine Auswahlfrage. Da er auch hier Befehle eingeben können soll, muß ein question frame gewählt werden, da im multiple choice frame vom Lernenden keine function aufgerufen werden kann (zum Frame-Aufbau in PLANIT s. Teil 4, Punkt 2.3.1.2.1.). Frame 2 generiert die "Beobachtungswerte" nach einem Polynom zweiter Ordnung und gibt sie in einer Tabelle aus. Will der Student eigene Daten eingeben, wird er in frame 3 nach der Anzahl der Werte gefragt, die er in den frames 4 bis 4.3 eingeben muß. In den frames 5 bis 10.5 kann er die Parameterwerte festlegen. In frame 11 wird das Verfahren durchgerechnet, dessen Ergebnisse in 11.1 tabellarisch ausgegeben werden. Der Rechenalgorithmus der exponentiellen Glättung ist in PLANIT fester Bestandteil des Dialogprogrammes. Er müßte also in ein anderes Lehrprogramm neu eingegeben werden, falls

1) Der Aufbau von PLANIT-Programmen wird in Teil 4, Punkt 2.3.1.2.1. erläutert.

er dort benötigt würde. Die Programmierung in einem als Unterprogramm benutzten Hauptprogramm ist äußerst schwierig (vgl. Teil 4, Punkt 2.3.1.2.1.). Kommt der Student zum ersten Mal zu frame 11.3, so erhält er eine Darstellung der Befehle (frame 11.4 und 11.5) und bekommt in frame 12 eine Auswahlfrage. Diese wird ihm bei jeder weiteren Durcharbeitung des Programmes sofort nach der Ergebnistabelle gestellt. Frame 13 schließt das Lernprogramm ab. Das Inhaltsverzeichnis dieses Kurses befindet sich in frame 11.6.

Die Eingabe der Beobachtungswerte hätte auch über eine function ermöglicht werden können, doch müßte sich der Lernende dann an ein bestimmtes Format halten. Es wurde daher die etwas aufwendigere aber sicherere Alternative verwirklicht.

Auf eine graphische Ausgabe wurde wegen des hohen Aufwandes für die Programmierung und wegen des hohen Rechenzeitbedarfes verzichtet. Zwar stünde mit dem align-Statement ein sehr leistungsfähiger Ausgabebefehl zur Verfügung, doch müßte der Maßstab errechnet und die Graphik um 90 Grad gedreht werden (vgl. Teil 4, Punkt 2.3.3.2.1.).

FRAME 1.00 (Q)

G2. TEXT
EXPONENTIELLE GLÄTTUNG

DRITTER ORDNUNG

FRAME 1.01 (Q)

G2. TEXT
Wir begrüßen Sie im Trainingsprogramm für exponentielle
Glättung dritter Ordnung. Sie können in diesem Kurs die
Eigenschaften dieses Prognoseverfahrens untersuchen, in-
dem Sie es an verschiedenen Datensets testen und die Pa-
rametereinstellungen modifizieren. Sie können auch den
Glättungszeitraum selbst bestimmen und so gezielte
Datenanalyse betreiben.@@@

FRAME 1.10 (Q)

G2. TEXT
Der exponentiellen Glättung dritter Ordnung liegt ein
quadratisches Modell folgender Art zugrunde:®
 $P(t) = a + b*t + c/2*t^2$
Legende:®
P(t).....Prozeßwert zum Zeitpunkt t
a.....Grundwert
b.....Trendwert
c.....Trendwert der Trendwerte®
Folgende Gleichungen werden zur Abschätzung der
Koeffizienten a, b und c verwendet:®
 $a(t) = x(t) + be**3*(xg(t-1) - x(t))$
 $b(t) = p(t-1) + c(t-1) - 3*al**2/2*(2-al)*(xg(t-1) - x(t))$
 $c(t) = c(t-1) - al**3*(xg(t-1) - x(t))$ ®
Legende:®
x(t).....Beobachtungswert zum Zeitpunkt t
xg(t+k)....Vorhersage zum Zeitpunkt t für k Perioden
al.....Glättungsparameter
be.....be=1-al®
Daraus ergibt sich folgende Vorhersagegleichung:®
 $xg(t+k) = a + k*b(t) + k**2/2*c(t)$.@@

FRAME 1.30 (P)

G2. STATEMENTS
l=1
nk=40
bef=1
set matrix(x,50)
set matrix(xg,50)
set matrix(al,50)
set matrix(b1,50)
set matrix(c1,50)
set matrix(aa,50)
function param=goto 5
function daten=goto 1.5
function eg3=goto 11
function inhalt=goto 11.6
function befehle=goto 11.4
function go(y)=goto y

FRAME 1.50 (Q)

G2. TEXT
Haben Sie®
a) eigene Daten, die Sie verwenden wollen, oder
b) wünschen Sie ein vom Rechner bereitgestelltes Datenset?

G3. ANSWERS
O allow all
O keyword 1
a a)
b b.)

G4. ACTIONS
a b: datein
- r:

FRAME 2.00 (P) LABEL=DATEN

G2. STATEMENTS

```
round 1
x(k)=100 + 5*random + k*2 + (k/10)**2 for k=1,40
print'Stellen Sie für folgende Umsatzdaten eine Prognose für'
print'50 Perioden auf.'
print'Beobachtungswerte'
align'Periode',1,'Periode',21
align'1-20',3,'21-40',23
align x(k),6; x(k+20),26 for k=1,20
nk = 40
b: 5
```

FRAME 3.00 (Q) LABEL=DATEIN

G2. TEXT

Wieviele Daten wollen Sie eingeben (zwischen 1 und 40)?

G3. ANSWERS

```
0 allow all
1 21 within 19
```

G4. ACTIONS

```
c: x(k) = 0 for k=1,50
1 c: nk=response
- r:
```

FRAME 4.00 (Q) LABEL=VIER

G2. TEXT

Geben Sie nun Ihre Daten ein!
x(1)=\$

G3. ANSWERS

```
0 allow all
1 0 within 10000
```

G4. ACTIONS

```
1 c: x(L)=response
c: L=2
```

FRAME 4.20 (D)

G2. CRITERIA

```
if L gr nk b: 5
else c: print 'x('; L round 0; ')=';$
```

FRAME 4.30 (Q)

G3. ANSWERS

```
0 allow all
1 0 within 100000
```

G4. ACTIONS

```
1 c: x(L)=response c: L= L+1 b: 4.2
```

FRAME 5.00 (Q)

G2. TEXT

Bitte bestimmen Sie nun die Parameter!
Glättungsparameter al=\$

G3. ANSWERS

```
0 allow all
1 0,5 within 0,5
```

G4. ACTIONS

```
1 c: al=response
- r:
```

FRAME 6.00 (Q) LABEL=A

G2. TEXT
Grundwert a=\$

G3. ANSWERS
0 allow all
1 0 within 100000

G4. ACTIONS
1 c: a=response
- r:

FRAME 7.00 (Q) LABEL=B

G2. TEXT
Trendwert b=\$

G3. ANSWERS
0 allow all
1 0 within 10000

G4. ACTIONS
1 c: b=response
- r:

FRAME 8.00 (Q) LABEL=C

G2. TEXT
Trendwert der Trendwerte c=\$

G3. ANSWERS
0 allow all
1 0 within 100

G4. ACTIONS
1 c: c=response
- r:

FRAME 9.00 (Q) LABEL=TANF

G2. TEXT
Ab welcher Periode wollen Sie glätten?
tanf=\$

G3. ANSWERS
0 allow all
1 25.5 within 24.5

G4. ACTIONS
1 c: tanf=truncate(response)
- r:

FRAME 10.00 (Q) LABEL=TEN

G2. TEXT
Bis zu welcher Periode soll geglättet werden?
tend=\$

G3. ANSWERS
0 allow all
1 25.5 within 24.5

G4. ACTIONS
1 c: tend=truncate(response)
- r:

FRAME 10.50 (D)

G2. CRITERIA
if tend > tanf f: tend muß größer als tanf sein!!! b: tan
else

FRAME 11,00 (P) LABEL=EG3

G2. STATEMENTS

```
ind=tanf-1
aaa=0
round 1
auch=nk
if tend qq nk b: lab7
else c: auch=tend
lab7: i=tanf-1
be = 1 - al
alcub=al**3
becub=be**3
a1(tanf)=a
b1(tanf)=b
c1(tanf)=c
lab8: i=i+1
if i gr auch b: lab
else c: xg(i)=a+b+0,5*c
dif=xg(i)-x(i)
a=x(i)-becub*dif
b=b+c-1,5*al*al*(2-al)*dif
c=c-alcub*dif
a1(i+1)=a
b1(i+1)=b
c1(i+1)=c
goto lab8
lab: if tend lq nk b: lab4
else c: id=nk
lab5: id=id+1
is=id-nk
xg(id)=a+b*is+0,5*c*is*is
if tend gr id b: lab5
else
lab4: aa(mm)=xg(mm)-x(mm) for mm= 1, tend
bbb: ind=ind+1
aaa=aaa+absolute(aa(ind))
if ind ls tend b: bbb
else
```

FRAME 11,10 (P)

G2. STATEMENTS

```
align 'x',5; 'xg',15; 'a',25; 'b',35; 'c',45; 'Abw:',55
auch=tend
if tanf lq 1 b: l1
else c: align x(k), 5 for k=1,tanf - 1
l1: if tend qq nk c: auch = nk b: l10
else b: l10
l10: align x(f),5; xg(f),15; a1(f),25; b1(f),35; c1(f),45; aa(f),56 for f=tanf,auch
if tend qq nk b: l4
else c: align x(k),5 for k= auch+1,nk b: l3
l4: if tend eq nk b: l3
else
align xg(k),15; a1(k),25;b1(k),35; c1(k),45 for k=auch + 1, tend
l3: a = a1(tanf)
b=b1(tanf)
c=c1(tanf)
aaa=aaa/(tend+1-tanf)
align 'Die mittlere absolute Abweichung beträgt: ', 1; aaa,46
```

FRAME 11,30 (D)

G2. CRITERIA

```
if bef nq 1 b: l2
else c: bef = 2
```

FRAME 11.40 (Q)

G2. TEXT

Im Rahmen dieses Kurses stehen Ihnen folgende Befehle zur Verfügung, die Sie an jedem Punkt des Programmes eingeben können:00
Befehl:0

<daten Sie können ein neues Datenset eingeben oder von der Maschine erstellen lassen.0
<param Sie können die Parameter neu bestimmen.0
<go(y) Mit diesem Befehl können Sie auf jedes beliebige Frame springen, das im Inhaltsverzeichnis aufgeführt wird (y ist die Nummer des gewünschten Frames).0
<inhalt Dieses steht Ihnen mit dem Befehl <inhalt zur Verfügung.0
<eg3 Geben Sie den Befehl <eg3 ein, wird das Verfahren der exponentiellen Glättung dritter Ordnung durchgerechnet.0

Parameter können auf folgende Art geändert werden:0

Beispiel:

<a=205

>

Folgende Parameter können Sie modifizieren:

a.....Glättungsparameter
a.....Grundwert
b.....Trendwert
c.....Trendwert der Trendwerte
tanf...Periode, ab der geglättet werden soll
tend...Periode, bis zu der geglättet werden soll0

Nach Eingabe des(r) Parameterwerte(s) geben Sie das Zeichen >.000

FRAME 11.50 (Q)

G2. TEXT

Vergessen Sie irgendwann, welche Befehle zur Verfügung stehen, so geben Sie

<befehle

ein und erhalten eine Liste der Kommandos.

G3. ANSWERS

0 allow all

G4. ACTIONS

b:12

FRAME 11.60 (Q)

G2. TEXT

Mit dem Befehl <go(y) können Sie folgende Punkte im Programm anspringen:

00

1.....Anfang
2.....Datenset, von Rechner bereitgestellt
3.....Eingabe eigener Daten
5.....Parametereingabe
11.5.....Befehlsliste
13.....Programmende

FRAME 12.00 (Q)

G2. TEXT

Wollen Sie num0

a) Parameter ändern,
b) ein anderes Beispiel wählen oder
c) das Training beenden?

G3. ANSWERS

0 allow all

0 keyword 1

a a a)

b b b)

c c c)

G4. ACTIONS

a b: 5

b b: 1,5

c b: stop

FRAME 13.00 (Q) LABEL=STOP

G2. TEXT

Sie sind nun am Ende des Trainingsprogrammes angelangt.

G4. ACTIONS

c: finished

2.1.3. Dialogbeispiel

Dialog:

Erklärung:

29.03.73 07:42:33 NR.49
IHRE PLANIT-KENNUNG BITTE

pl
ENTER COMMAND
*...
IDENTIFY YOURSELF***pl
ENTER COMMAND
*... 1

EXPONENTIELLE GLAETTUNG

DRITTER ORDNUNG

Wir begrüßen Sie im Trainingsprogramm für exponentielle Glättung dritter Ordnung. Sie können in diesem Kurs die Eigenschaften dieses Prognoseverfahrens untersuchen, indem Sie es an verschiedenen Datensets testen und die Parameter Einstellungen modifizieren. Sie können auch den Glättungszeitraum selbst bestimmen und so gezielte Datenanalyse betreiben.

A 1

Der exponentiellen Glättung dritter Ordnung liegt ein quadratisches Modell folgender Art zugrunde:

$$P(t) = a + b \cdot t + c/2 \cdot t^2$$

Legende:

P(t).....Prozeßwert zum Zeitpunkt t
a.....Grundwert
b.....Trendwert
c.....Trendwert der Trendwerte

Folgende Gleichungen werden zur Abschätzung der Koeffizienten a, b und c verwendet:

$$\begin{aligned} a(t) &= x(t) + be**3*(xg(t-1) - x(t)) \\ b(t) &= b(t-1) + c(t-1) - 3*aL**2/2*(2-al)*(xg(t-1) - x(t)) \\ c(t) &= c(t-1) - al**3*(xg(t-1) - x(t)) \end{aligned}$$

Legende:

x(t).....Beobachtungswert zum Zeitpunkt t
xg(t+k)....Vorhersage zum Zeitpunkt t für k Perioden
al.....Glättungsparameter
be.....be=1-al

Daraus ergibt sich folgende Vorhersagegleichung:

$$xg(t+k) = a + k*b(t) + k**2/2*c(t).$$

Da der Student bereits vor dem Training mit dem Verfahren vertraut sein muß, genügt hier zur Auffrischung des Gedächtnisses eine knappe Darstellung des Modells.

A 2

Haben Sie

- a) eigene Daten, die Sie verwenden wollen, oder
b) wünschen Sie ein vom Rechner bereitgestelltes Datenset?

A 3

*- Stellen Sie für folgende Umsatzdaten eine Prognose für 50 Perioden auf.
Beobachtungswerte

Periode	Periode
1-20	21-40
105,4	149,0
109,5	152,3
106,9	153,5
109,3	155,8
111,6	156,3
115,1	163,4
117,2	165,2
120,9	165,4
120,2	167,3
125,2	170,9
125,7	171,7
126,0	175,5
127,9	177,5
132,3	183,1
134,0	184,4
135,1	189,5
139,9	190,0
143,2	193,8
145,6	195,2
147,4	199,5

Das Programm stellt dem Studenten auf Wunsch ein zufallsgeneriertes Datenset zur Verfügung. Der Student kann somit mehrere unterschiedliche Beispiele bearbeiten.

A 4

Bitte bestimmen Sie nun die Parameter!
Glättungsparameter $\alpha = 0,3$

Grundwert $a = 102$

Trendwert $b = 3$

Trendwert der Trendwerte $c = 1$

Ab welcher Periode wollen Sie glätten?
 $t_{anf} = 1$

Bis zu welcher Periode soll geglättet werden?
 $t_{end} = 10$

A 5
A 6
A 7
A 8
A 9
A 10
A 11

x	x_0	a	b	c	Abw.
105,4	105,0	102,0	3,0	0,1	-0,3
109,5	109,7	105,5	3,2	0,1	0,2
106,9	111,7	109,4	3,2	0,1	4,8
109,3	107,5	105,2	2,2	0,0	-0,8
111,6	111,0	109,6	2,4	0,0	-0,6
115,1	114,3	111,8	2,5	0,0	-0,8
117,2	119,2	115,4	2,7	0,0	1,0
120,9	119,4	116,8	2,5	0,0	-1,4
120,2	124,1	121,2	2,9	0,0	3,9
125,2	120,5	118,3	2,0	-0,1	-4,4
125,7					
126,0					
127,9					
132,3					
134,0					
135,1					
139,9					
143,2					
145,6					
147,4					
149,0					
152,3					
153,5					
155,8					
156,3					
163,4					
165,2					
165,4					
167,3					
170,9					
171,7					
175,5					
177,5					
183,1					
184,4					
189,5					
190,0					
193,8					
195,2					
199,5					

Dem Studenten wird neben den im Programm errechneten Werten die Reihe der Beobachtungswerte vollständig ausgegeben. Er kann sich daher bereits beim Studium der Tabelle überlegen, ob sein Modell auch für weitere Perioden stimmen könnte.

A 12

Die mittlere absolute Abweichung beträgt: 1,8

Im Rahmen dieses Kurses stehen Ihnen folgende Befehle zur Verfügung, die Sie an jedem Punkt des Programmes eingeben können:

Befehl:

- <daten Sie können ein neues Datenset eingeben oder von der Maschine erstellen lassen.
- <param Sie können die Parameter neu bestimmen.
- <go(y) Mit diesem Befehl können Sie auf jedes beliebige Frame springen, das im Inhaltsverzeichnis aufgeführt wird (y ist die Nummer des gewünschten Frames).
- <inhalt Dieses steht Ihnen mit dem Befehl <inhalt zur Verfügung.
- <eg3 Geben Sie den Befehl <eg3 ein, wird das Verfahren der exponentiellen Glättung dritter Ordnung durchgeführt.

Parameter können auf folgende Art geändert werden:

Beispiel:
<a=205
>

Folgende Parameter können Sie modifizieren:

- a.....Glättungsparameter
- b.....Grundwert
- b.....Trendwert
- c.....Trendwert der Trendwerte
- tanf....Periode, ab der geglättet werden soll
- tend....Periode, bis zu der geglättet werden soll

Nach Eingabe des(r) Parameterwerte(s) geben Sie das Zeichen >.

Vergessen Sie irgendwann, welche Befehle zur Verfügung stehen, so geben Sie

<befehle
ein und erhalten eine Liste der Kommandos.

Bei der ersten Durcharbeitung werden dem Lernenden alle verfügbaren Befehle erklärt. Die hier angebotenen Kommandos bringen in diesem kleinen Versuchsbeispiel zum Teil wenig Nutzen. Sie wurden primär zur Untersuchung geschaffen, ob und wie Befehle in PLANIT vom Autor definiert werden können.

} A 13

Wollen Sie nun

- a) Parameter ändern,
- b) ein anderes Beispiel wählen oder
- c) das Training beenden?

} A 14

Bitte bestimmen Sie nun die Parameter!
Glättungsparameter $\alpha = 0.6$

Grundwert $a = 102$

Trendwert $b = 3$

Trendwert der Trendwerte $c = 1$

Ab welcher Periode wollen Sie glätten?
 $\text{tanf} = 1$

Sis zu welcher Periode soll geglättet werden?
 $\text{tend} = 10$

x	x _g	a	b	c	Abw.
105,4	105,0	102,0	3,0	0,1	-0,3
109,5	108,9	105,4	3,4	0,2	0,4
106,9	111,8	108,5	3,3	0,1	4,9
109,3	105,8	106,6	-0,3	-1,0	-2,5
111,6	108,9	108,5	0,6	-0,4	-2,7
115,1	114,1	111,8	2,3	0,2	-1,0
117,2	118,6	115,2	3,2	0,4	1,5
120,8	119,6	117,1	2,5	0,1	-1,1
120,2	124,4	120,8	3,4	0,3	4,3
125,2	120,1	119,9	0,5	-0,6	-5,0
125,7					
126,0					
127,9					
132,3					
134,0					
135,1					
139,9					
143,2					
145,6					
147,4					
149,0					
152,3					
153,5					
155,8					
156,3					
163,4					
165,2					
165,4					
167,3					
170,9					
171,7					
175,5					
177,5					
183,1					
184,4					
189,5					
190,0					
193,3					
195,2					
199,5					

Die mittlere absolute Abweichung beträgt: 2,4

Wollen Sie nun

- a) Parameter ändern,
- b) ein anderes Beispiel wählen oder
- c) das Training beenden?

*<nl>=2
AL = 0,2

*tend=11
TANF = 11,0

*tend=20
TEND = 20,0

*t=117,3
A = 118,8

*t=7
B = 2,0

*t=-1
C = -0,1

*en3	x	xg	a	b	c
105,4					
106,5					
106,9					
108,3					
111,6					
115,1					
117,2					
120,3					
120,2					
125,2					
125,7	120,7	118,8	2,0	-0,1	-5,0
127,0	130,7	128,3	2,4	-0,1	4,7
127,9	125,4	123,6	1,9	-0,1	-2,5
132,3	131,2	129,2	2,0	-0,1	-1,1
134,0	134,9	132,9	2,1	-0,1	1,0
135,1	135,3	133,5	1,9	-0,1	0,2
139,9	136,7	135,0	1,9	-0,1	-3,1
143,2	143,5	141,4	2,1	-0,1	0,2
145,6	145,1	143,1	2,0	-0,1	-0,5
147,4	147,3	145,8	2,0	-0,1	0,3
149,0					
152,3					
153,5					
155,8					
156,3					
163,4					
165,2					
165,4					
167,3					
170,9					
171,7					
175,5					
177,5					
183,1					
184,4					
199,5					
199,5					
199,5					
199,5					

Die mittlere absolute Abweichung beträgt: 1,9

Da der Student die Befehle nur dann einsetzen soll, wenn er vom vorgesehenen Weg abweichen will, werden ihm verschiedene Fortsetzungsalternativen angeboten. Er muß aber auf diese Frage, wie das Beispiel zeigt, nicht antworten, sondern kann mit Befehlen den Dialog steuern. Hat der Student einmal das Zeichen < eingegeben, so muß er dies bei den folgenden Befehlseingaben nicht wiederholen.

Abw. Der Student untersucht nun, ob sein Modell, das er für die ersten zehn Beobachtungswerte gebildet hat, auch für die zweiten zehn gilt.

Wollen Sie nun

- a) Parameter ändern,
- b) ein anderes Beispiel wählen oder
- c) das Training beenden?

*<tanf=1
TANF = 1.0.

*<tend=30
TEND = 30.0

*a=102
A = 102.0

*b=3
B = 3.0

*c=.1
C = 0.1

*eg3

x	xg	a	b	c	Abw.
105.4	105.0	102.0	3.0	0.1	-0.3
108.5	108.8	105.6	3.1	0.1	0.3
106.9	111.6	108.4	3.2	0.1	4.7
108.3	107.3	104.5	2.8	0.1	-1.0
111.6	111.8	108.8	3.0	0.1	0.2
115.1	114.5	111.5	3.0	0.1	-0.6
117.2	118.6	115.5	3.2	0.1	1.5
120.8	119.5	116.4	3.1	0.1	-1.2
120.2	124.7	121.4	3.3	0.1	4.5
125.2	120.7	117.9	2.8	0.0	-4.5
125.7	130.9	127.5	3.4	0.1	5.1
126.0	126.0	123.1	2.9	0.0	0.0
127.9	128.9	126.0	2.9	0.0	1.0
132.3	130.2	127.4	2.8	0.0	-2.1
134.0	136.5	133.4	3.1	0.0	2.5
135.1	135.5	132.7	2.8	0.0	0.4
136.9	137.7	134.9	2.8	0.0	-2.2
143.2	144.0	141.0	3.1	0.0	0.8
145.6	145.9	142.8	3.0	0.0	0.3
147.4	148.4	145.4	3.0	0.0	1.0
149.0	149.9	146.9	2.9	0.0	0.9
152.3	151.4	148.5	2.8	0.0	-0.9
153.5	155.8	152.8	3.0	0.0	2.3
155.3	155.0	152.3	2.7	0.0	-0.8
156.3	159.0	156.2	2.8	0.0	2.7
163.4	157.4	154.9	2.5	0.0	-5.9
165.2	169.6	166.4	3.1	0.0	4.3
165.4	165.7	163.0	2.7	0.0	0.3
167.3	167.9	165.2	2.7	0.0	0.6
170.9	169.6	167.0	2.6	0.0	-1.3
171.7					
175.5					
177.5					
183.1					
184.4					
189.5					
190.0					
193.3					
195.2					
199.5					

Die mittlere absolute Abweichung beträgt: 1.8

Wollen Sie nun

- a) Parameter ändern,
- b) ein anderes Beispiel wählen oder
- c) das Training beenden?

*al=.3

AL = 0.3

*tend=45

TEND = 45.0

*eg3

x	xg	a	b	c	Abw.
105.4	105.0	102.0	3.0	0.1	-0.3
103.5	103.7	105.5	3.2	0.1	0.2
106.9	111.7	108.4	3.2	0.1	4.8
108.3	107.5	105.2	2.2	0.0	-0.8
111.6	111.0	108.6	2.4	0.0	-0.6
115.1	114.3	111.8	2.5	0.0	-0.8
117.2	118.2	115.4	2.7	0.0	1.0
120.8	119.4	116.8	2.5	0.0	-1.4
120.2	124.1	121.2	2.9	0.0	3.9
125.2	120.8	118.8	2.0	-0.1	-4.4
125.7	129.7	126.7	2.9	0.1	3.9
126.0	126.5	124.4	2.1	0.0	0.5
127.9	127.7	125.8	1.9	-0.1	-0.2
132.3	129.8	128.0	1.9	-0.1	-2.5
134.0	135.6	133.1	2.4	0.0	1.6
135.1	135.5	133.4	2.1	0.0	0.3
139.9	136.9	135.0	1.9	0.0	-3.0
143.2	143.5	140.9	2.6	0.0	0.2
145.6	145.7	143.2	2.6	0.0	0.2
147.4	148.0	145.5	2.5	0.0	0.6
149.0	149.7	147.2	2.4	0.0	0.7
152.3	151.0	148.8	2.3	0.0	-1.3
153.5	155.4	152.8	2.6	0.0	1.9
155.8	154.9	152.8	2.2	0.0	-0.8
156.3	158.4	156.1	2.3	0.0	2.1
163.4	157.4	155.6	1.8	-0.1	-6.0
165.2	168.6	165.4	3.1	0.1	3.4
165.4	166.6	164.1	2.5	0.0	1.2
167.3	167.2	165.0	2.2	0.0	-0.1
170.9	169.5	167.3	2.2	0.0	-1.4
171.7	173.9	171.4	2.5	0.0	2.2
175.5	172.9	170.9	2.0	0.0	-2.6
177.5	178.9	176.3	2.6	0.0	1.5
183.1	179.2	177.0	2.3	0.0	-3.9
184.4	187.6	184.4	3.1	0.1	3.2
189.5	185.8	183.3	2.5	0.0	-3.7
190.0	194.2	190.8	3.3	0.1	4.1
193.8	191.1	188.6	2.5	0.0	-2.7
195.2	197.9	194.8	3.1	0.1	2.7
199.5	196.8	194.3	2.6	0.0	-2.6
	203.6	200.4	3.2	0.1	
	206.8				
	210.2				
	213.6				
	217.0				

Die mittlere absolute Abweichung beträgt: 2.0

Nun hat der Student eine Prognose für fünf Perioden erstellt.

Wollen Sie nun

- a) Parameter ändern,
- b) ein anderes Beispiel wählen oder
- c) das Training beenden?

*<befehle

Im Rahmen dieses Kurses stehen Ihnen folgende Befehle zur Verfügung, die Sie an jedem Punkt des Programmes eingeben können:

Befehl:

- <daten Sie können ein neues Datenset eingeben oder von der Maschine erstellen lassen.
- <param Sie können die Parameter neu bestimmen.
- <go(y) Mit diesem Befehl können Sie auf jedes beliebige Frame springen, das im Inhaltsverzeichnis aufgeführt wird (y ist die Nummer des gewünschten Frames).
- <inhalt Dieses steht Ihnen mit dem Befehl <inhalt zur Verfügung.
- <eg3 Geben Sie den Befehl <eg3 ein, wird das Verfahren der exponentiellen Glättung *drift* der Ordnung durchgeführt.

Parameter können auf folgende Art geändert werden:

Beispiel:

<a=205
>

Folgende Parameter können Sie modifizieren:

- 3l.....Glättungsparameter
- a.....Grundwert
- b.....Trendwert
- c.....Trendwert der Trendwerte
- tanf...Periode, ab der geglättet werden soll
- tend...Periode, bis zu der geglättet werden soll

Nach Eingabe des(r) Parameterwerte(s) geben Sie das Zeichen >.

Vergessen Sie irgendwann, welche Befehle zur Verfügung stehen, so geben Sie

<befehle

ein und erhalten eine Liste der Kommandos.

Wollen Sie nun

- a) Parameter ändern,
- b) ein anderes Beispiel wählen oder
- c) das Training beenden?

*<inhalt

Mit dem Befehl <go(y) können Sie folgende Punkte im Programm anspringen:

- 1.....Anfang
- 2.....Datenset, von Rechner bereitgestellt
- 3.....Eingabe eigener Daten
- 5.....Parametereingabe
- 11.5.....Befehlsliste
- 13.....Programmende

Wollen Sie nun

- a) Parameter ändern,
- b) ein anderes Beispiel wählen oder
- c) das Training beenden?

*<go(3)

Der Lernende weiß nicht mehr, welche Befehle vorhanden sind, und läßt sich eine Liste ausgeben.

Wieviele Daten wollen Sie eingeben (zwischen 1 und 40)? } A 15

*24

Geben Sie nun Ihre Daten ein! } A 16

```
x(1)=222.3
x(2)=220.7
x(3)=219.1
x(4)=217.4
x(5)=215.4
x(6)=214.0
x(7)=211.3
x(8)=209.7
x(9)=207.2
x(10)=203.5
x(11)=203.1
x(12)=200.0
x(13)=197.7
x(14)=194.0
x(15)=194.7
x(16)=194.0
x(17)=194.0
x(18)=195.2
x(19)=197.2^3
x(20)=191.0
x(21)=194.3
x(22)=194.7
x(23)=194.5^2.5
x(24)=243.3
```

Bitte bestimmen Sie nun die Parameter!
Glättungsparameter al= 0.4

Grundwert a=224

Trendwert b=1

Trendwert der Trendwerte c=.2

Ab welcher Periode wollen Sie glätten?
tanf=1

Bis zu welcher Periode soll geglättet werden?
tend=12

x	xg	a	b	c	Abw.
222.3	225.1	224.0	1.0	0.2	2.8
220.7	221.8	221.7	0.1	0.0	1.1
219.1	220.1	220.5	-0.3	-0.1	1.0
217.4	218.1	218.9	-0.7	-0.1	-1.3
215.4	219.3	219.7	-0.3	0.0	-0.1
214.0	219.1	219.4	-0.4	0.0	-4.9
211.3	226.7	225.1	1.5	0.3	-5.1
209.7	237.0	232.9	3.8	0.6	1.3
207.2	239.6	235.4	3.9	0.5	2.4
203.5	240.4	236.7	3.5	0.4	1.9
203.1	241.4	238.1	3.2	0.3	3.3
200.0	239.6	237.4	2.2	0.0	2.7
197.7					
194.0					
194.7					
194.0					
195.2					
197.2^3					
191.0					
194.3					
194.7					
194.5^2.5					
243.3					

Die mittlere absolute Abweichung beträgt: 2.3

Diese Art der Dateneingabe ist verhältnismäßig mühsam. Sie bietet jedoch gegenüber der Möglichkeit der Eingabe mit function den Vorteil der Formatfreiheit.

Wollen Sie nun

- a) Parameter ändern,
- b) ein anderes Beispiel wählen oder
- c) das Training beenden?

*a

Bitte bestimmen Sie nun die Parameter!
Glättungsparameter $\alpha = .4$

Grundwert $a=224$

Trendwert $b=-1$

Trendwert der Trendwerte $c=.1$

Ab welcher Periode wollen Sie glätten?
 $\text{tanf}=1$

Bis zu welcher Periode soll geglättet werden?
 $\text{tend}=12$

x	xg	a	b	c	Abw.
222.3	223.0	224.0	-1.0	0.1	0.3
220.7	221.0	222.1	-1.2	0.1	0.3
219.1	219.4	220.6	-1.2	0.0	0.3
219.4	217.7	219.0	-1.3	0.0	-1.7
219.4	219.2	219.8	-0.7	0.1	-0.2
224.0	219.1	219.5	-0.5	0.1	-4.9
231.8	226.9	225.1	1.6	0.5	-4.9
235.7	227.2	232.9	3.9	0.8	1.5
237.2	239.8	235.4	4.1	0.7	2.6
239.5	240.7	236.6	3.8	0.5	2.2
239.1	241.7	238.0	3.5	0.4	3.6
236.9	239.8	237.3	2.5	0.1	2.9
234.7					
234.0					
233.2					
235.2					
238.3					
241.0					
244.8					
249.9					
251.8					
251.7					
253.5					
249.3					

Die mittlere absolute Abweichung beträgt: 2.0

Wollen Sie nun

- a) Parameter ändern,
- b) ein anderes Beispiel wählen oder
- c) das Training beenden?

*a

Bitte bestimmen Sie nun die Parameter!
Glättungsparameter $\alpha = .6$

Grundwert $a=223.2^{***}38.0$

Trendwert $b=0$

Trendwert der Trendwerte $c=-.9$

Ab welcher Periode wollen Sie glätten?
 $\text{tanf}=13$

Bis zu welcher Periode soll geglättet werden?
 $\text{tend}=18$

x	xg	a	b	c	Abw.
222.3					
220.7					
219.1					
219.4					
219.4					
224.0					
231.8					
235.7					
237.2					
238.5					
239.1					
236.9					
234.7	237.5	239.0	0.0	-0.9	2.8
234.0	230.7	234.5	-3.1	-1.5	-3.3
233.2	231.7	234.2	-2.1	-0.8	-1.5
235.2	231.3	233.3	-1.8	-0.5	-3.9
238.3	236.3	235.5	0.7	0.4	-2.0
241.0	241.4	238.4	2.6	0.8	0.4
244.8					
249.9					
251.8					
251.7					
253.5					
249.3					

Die mittlere absolute Abweichung beträgt: 2.3

Wollen Sie nun

- a) Parameter ändern,
- b) ein anderes Beispiel wählen oder
- c) das Training beenden?

*al=7
AL = 0.7

x	xg	a	b	c	Abw.
222.3					
220.7					
219.1					
219.4					
219.4					
224.0					
231.8					
235.7					
237.2					
238.5					
233.1					
236.9					
234.7	237.5	239.0	0.0	-0.9	2.8
234.0	230.1	234.6	-3.6	-1.9	-3.9
233.2	232.1	234.1	-1.7	-0.5	-1.1
235.2	231.9	233.2	-1.2	-0.2	-3.3
238.3	237.5	235.3	1.7	1.0	-0.8
241.0	242.4	238.3	3.5	1.2	1.4
244.8					
249.9					
251.8					
251.7					
253.5					
249.3					

Die mittlere absolute Abweichung beträgt: 2.2

Wollen Sie nun

- a) Parameter ändern,
- b) ein anderes Beispiel wählen oder
- c) das Training beenden?

```
*<tarf=1
TANF = 1.0

*trnd=30
TEND = 30.0

*a=223.2
A = 223.2

*b=-1.
B = -1.0

*c=.1-
C = 0.1
```

```
*eg3
x      xg      a      b      c      Abw:
222.3  222.2  223.2  -1.0  0.1  -0.1
220.7  221.5  222.3  -0.9  0.1  0.8
219.1  219.1  220.7  -1.5  -0.2  0.0
219.4  217.4  219.1  -1.7  -0.2  -2.0
219.4  219.9  219.5  0.1  0.5  0.5
224.0  219.8  219.4  0.2  0.4  -4.2
231.8  229.6  224.1  4.6  1.8  -2.2
235.7  241.6  231.9  8.5  2.6  5.9
237.2  241.2  235.5  5.4  0.5  4.0
238.5  238.8  237.1  2.1  -0.8  0.3
238.1  239.0  238.5  1.0  -0.9  0.9
236.9  236.6  238.1  -0.8  -1.3  -0.3
234.7  234.5  236.9  -1.8  -1.2  -0.2
234.0  231.4  234.7  -2.8  -1.1  -2.6
233.2  232.6  234.1  -1.4  -0.2  -0.6
235.2  232.2  233.2  -1.0  0.0  -3.0
232.3  237.7  235.3  1.9  1.0  -0.6
241.0  242.5  236.3  3.5  1.3  1.5
244.8  244.7  241.0  3.4  0.8  -0.1
242.9  249.4  244.8  4.2  0.8  -0.5
251.8  255.9  249.9  5.5  1.0  4.1
251.7  254.0  251.7  2.5  -0.4  2.3
253.5  250.9  251.6  -0.1  -1.2  -2.6
249.3  254.5  253.6  1.1  -0.3  5.2
      243.9  249.2  -4.2  -2.1
      236.5
      226.9
      215.2
      201.4
      185.4
```

Die mittlere absolute Abweichung beträgt: 1.8

Wollen Sie nun

- a) Parameter ändern,
- b) ein anderes Beispiel wählen oder
- c) das Training beenden?

*c

Sie sind nun am Ende des Trainingsprogrammes angelangt:
 LESSON END
 ENTER COMMAND

2.1.4. Zusammenfassung

Das Experiment ist in wesentlichen Teilen durch die Tatsache bestimmt, daß PLANIT für die Realisation von RPU konzipiert wurde. So ist beispielsweise die Dokumentation der Lehrprogramme verhältnismäßig gut, wenn die Frame-Struktur für das Problem, das programmiert werden soll, geeignet ist. In unserem Experiment war der Frame-Aufbau in einem großen Teil des Programmes eher hinderlich. Dies zeigen die häufig auftretenden "verkümmerten" Frames deutlich.

Aber nicht nur die Dokumentation, sondern noch viel mehr die Programmierung wird durch den vorgegebenen Aufbau von PLANIT-Programmen behindert (s. z.B. frame 3 bis 4.3). Die Erstellung von Rechenroutinen in programming-frames ist jedoch kaum aufwendiger als in FORTRAN.

Die Antwortzeiten im Dialog sind meist günstig (zw. 1 und 4 sec.). Tritt jedoch irgendeine ungünstige Konstellation der Rechnerauslastung auf, werden diese beträchtlich höher. Sehr hoch sind die Reaktionszeiten des Rechners bei umfangreichen Berechnungen. Bei der Durchrechnung der exponentiellen Glättung dritter Ordnung für 40 Perioden wurden von uns Antwortzeiten zwischen 1 min. 30 sec. und 7 min. 50 sec. bei einem Mittelwert von 4 min. 50 sec. gemessen. Bedenkt man, daß der Rechner den Dialogbenutzern Zeitscheiben von 3 sec. zuteilt, so ist diese Antwortzeit nicht nur aus didaktischen Gründen dem Lernenden kaum zumutbar, sondern auch aus Wirtschaftlichkeitsgründen (Rechenzeitkosten) schwer vertretbar.

2.2. PUMA

Das System PUMA stand uns auf der CD 3300¹⁾ des Rechenzentrums der Universität Erlangen-Nürnberg, auf der auch das

1) CD steht für Control Data.

TS erstellt wurde, zur Verfügung. PUMA ist zwar zur kostengünstigen Realisation von RPU entwickelt worden, erlaubt jedoch einen wesentlich freieren Aufbau der Lehrprogramme als beispielsweise PLANIT. Da es aber trotzdem nicht möglich ist, das CET im geplanten Umfang in diesem System zu realisieren¹⁾, wird es im Projekt CET hauptsächlich für RPU und Selbsttests eingesetzt²⁾. Das Programm, das wir hier besprechen, wurde daher aus einem PU-Teil herausgegriffen, in dem primär untersucht werden sollte, wie benutzerfreundlich PUMA für derartige Strategien ist.

Das flexible Konzept von PUMA gestattet es, das ursprüngliche System um wesentliche Teile zu erweitern³⁾, die es nun erlauben, kleinere Trainingsprogramme zu verwirklichen. Die compilierende Verarbeitungsart ermöglicht zudem eine sehr schnelle Berechnung von Algorithmen. Um auch diese Anwendungsart hier zu zeigen, wird im Anschluß an das PU-Programm, das in das Verfahren der Schrittweisen Regression mit adaptiver Einflußgrößenkombination einführen soll, eine Routine zur Berechnung der exponentiellen Glättung dritter Ordnung dargestellt.

2.2.1. Didaktogramm

● Dialogablauf

Siehe Abb. 4.

● Befehle

Folgende vom Autor eingebauten Befehle stehen dem Studenten an einigen Stellen des Dialogablaufes zur Verfügung:

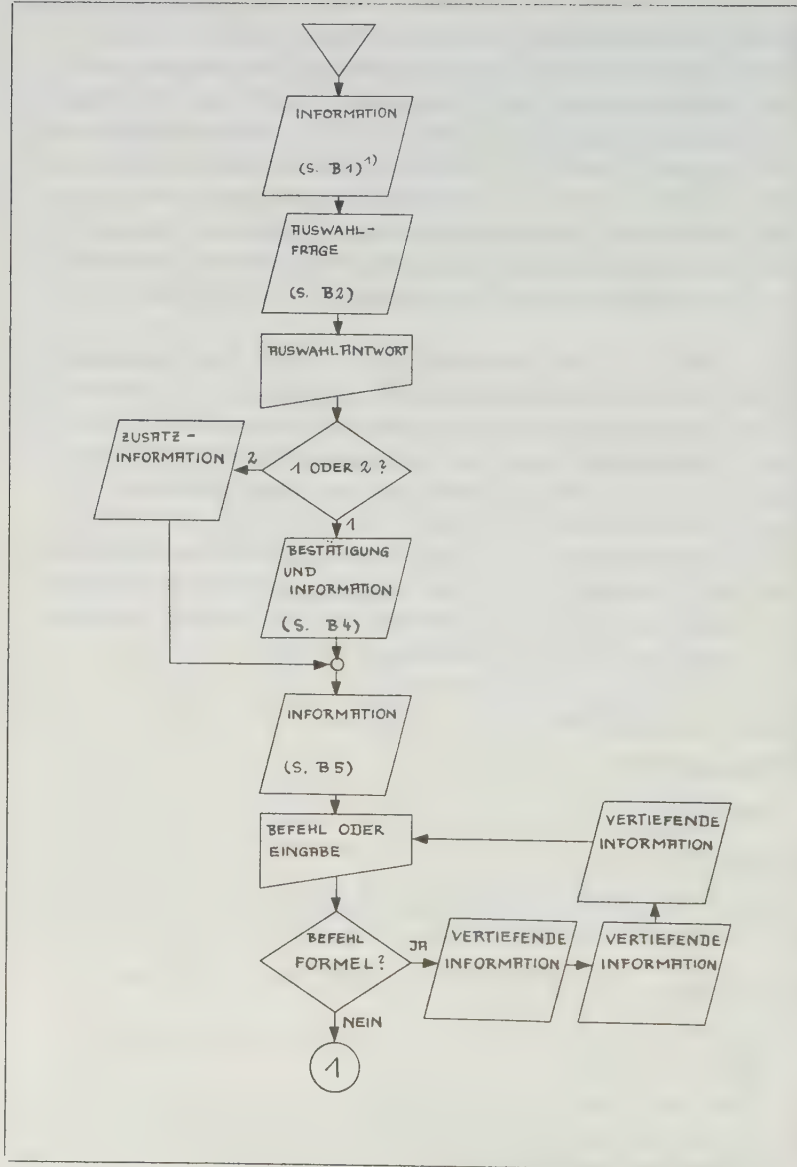
- FORMEL..... Der Lernende kann eine formale Erklärung für

1) Vgl. die Beschreibung der Realisierungsmöglichkeiten in PUMA im Teil 4.

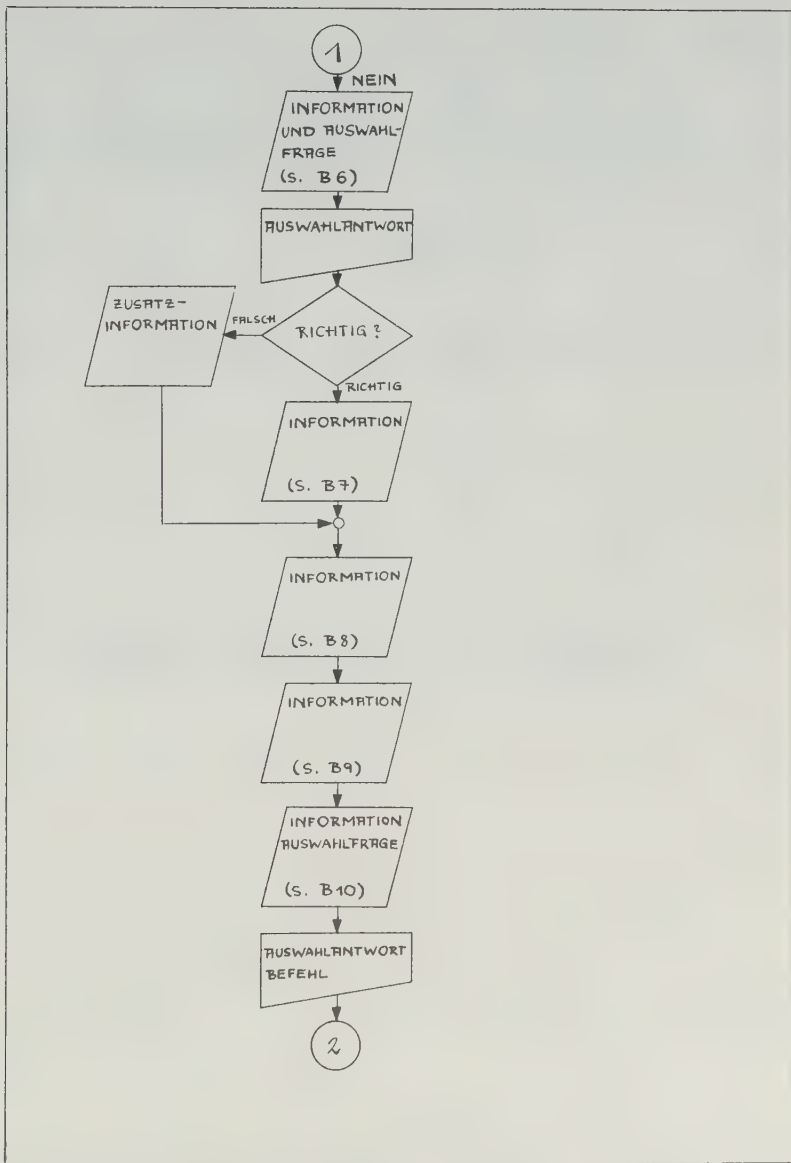
2) Über den Einsatz von PUMA berichtet G. Holub im Rahmen einer Dissertation genauer. Vgl. Holub, G., Probleme beim Einsatz des Computerunterstützten Entscheidungstrainings in der betriebswirtschaftlichen Ausbildung, Dissertation (in Arbeit) Nürnberg.

3) Vgl. Endres, C., a.a.O., S. 2.

Dialogablauf:



1) B i ist ein Bild, das im Dialogbeispiel gezeigt wird.



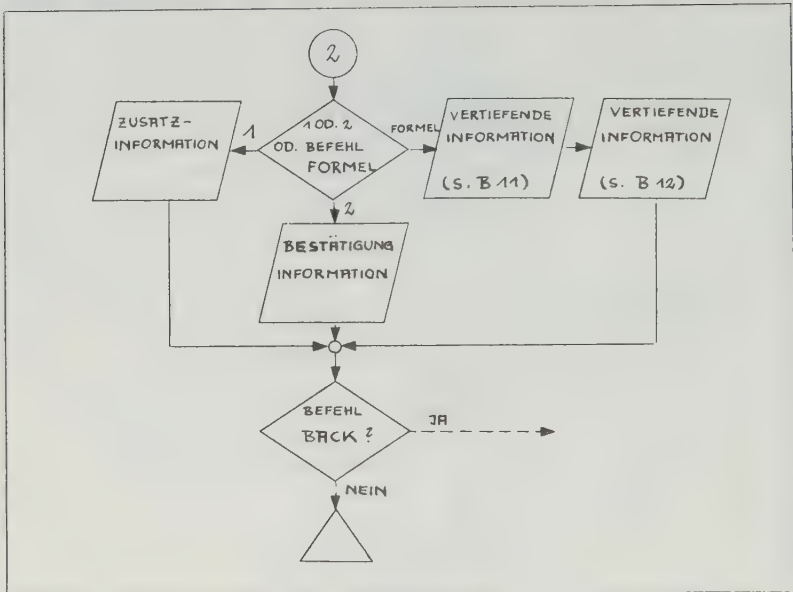


Abb. 4: Auszug aus dem Didaktogramm für das Lehrprogramm "Schrittweise Regression mit adaptiver Einflußgrößenkombination"

bestimmte Abschnitte des Programmes verlangen.

- BACK..... Mit dem Befehl BACK kann er auf einen früheren Punkt im Dialog zurückspringen.

Daneben hat er Zugriff auf alle PUMA-Befehle (vgl. Teil 4, Punkt 2.3.5.2.2.).

2.2.2. Programm

Der unten dargestellte Programmteil stammt aus einem Lehrprogramm über die Schrittweise Regression mit adaptiver Einflußgrößenkombination.

Im ersten Statement des Knotens A039 wird ein Zähler hochgesetzt, der zur schülerindividuellen Zeitmessung bestimmt ist. In Zeile 002 dieses Knotens wird mit einem DP-(Display-)Befehl das Bild mit dem Namen AT400 ausgegeben (zur Textausgabe s. Teil 4, Punkt 2.3.3.2.2.). Anschließend geht das Programm weiter auf Knoten A400 (zur Gliederung der PUMA-Programme in Knoten s. Teil 4, Punkt 2.3.1.2.2.). Auf die Parameter, die in der DP-Anweisung neben dem Folgeknoten und einem oder zwei Textkonservennamen angegeben werden können, wird hier nicht eingegangen, da sie für das Experiment unbedeutend sind.

In Zeile 002 des Knotens A040 wird der Inhalt der ersten Zahl, die vom Studenten eingegeben wurde, auf X1 gespeichert. Der weitere Ablauf innerhalb des Knotens ist ähnlich wie in FORTRAN. Zeile 002 des Knotens A043 beinhaltet einen Exaktvergleich ohne Berücksichtigung von Blanks. Im PUMA gibt es zudem den Exaktvergleich mit Berücksichtigung von Blanks und den Stichwortvergleich, die jedoch hier nicht verwendet werden.

Der Knoten A043 zeigt, wie in PUMA Befehle verarbeitet werden können. Soll der Student einen Befehl an jedem Punkt im Dialogablauf eingeben können, so muß in jedem Knoten danach abgefragt werden (vgl. Teil 4, Punkt 2.3.5.2.2.).

```

      KNOTEN A039
001      X100=X100+TIME(2)
002      DP(A000,AT000,,,,,1)

      KNOTEN A400
001      X100=X100+TIME(2)
002      DP(A040,AT040,,,,,1)

      KNOTEN A040
001      X100=X100+TIME(2)
002      X1=NUMBER(1)
003      X1=X1+1
004      X2=NUMBER(2)
005      IF(X2)5,1,1
006      5      GOTO(1,2,3,1)X1
007      1      DP(A040,AT999,AT040,,,,,1)
008      2      DP(A042,AT042,,,,,1)
009      3      DP(A042,AT)41,,,,,1)

      KNOTEN A542
001      X100=X100+TIME(2)
002      DP(A043,AT043,,,,,1)

      KNOTEN A043
001      X100=X100+TIME(2)
002      X1=EQUAL(IFORMEL#)
003      IF(X1)1,1,2
004      1      DP(A045,AT045,,,,,1)
005      2      DP(A070,AT070,,,,,1)

      KNOTEN A070
001      X100=X100+TIME(2)
002      DP(A071,AT071,,,,,1)

      KNOTEN A071
001      X100=X100+TIME(2)
002      DP(A072,AT072,,,,,1)

      KNOTEN A072
001      X100=X100+TIME(2)
002      DP(A043,AT043,,,,,1)

```

KNOTEN A045

```

001      X100=X100+TIME(2)
002      X1=NUMBER(1)
003      X1=X1+1
004      X2=NUMBER(2)
005      IF(X2)5,1,1
006      5      GOTO(1,2,3,1)X1
007      1      DP(A045,AT339,AT045,,,1)
008      2      DP(AJ46,ATJ46,,,1)
009      3      DP(A046,ATJ47,,,1)

```

KNOTEN A046

```

001      X100=X100+TIME(2)
002      DP(A490,ATJ48,,,1)

```

KNOTEN A490

```

001      X100=X100+TIME(2)
002      DP(A049,ATJ49,,,1)

```

KNOTEN A073

```

001      X100=X100+TIME(2)
002      DP(A124,AT074,,,1)

```

KNOTEN A074

```

001      X100=X100+TIME(2)
002      DP(A124,AT051,,,1)

```

KNOTEN A048

```

001      X100=X100+TIME(2)
002      DP(A049,AT490,,,1)

```

KNOTEN A049

```

001      X100=X100+TIME(2)
002      X1=EQUAL(#FORMEL#)
003      X2=EQUAL(#1#)
004      X3=EQUAL(#2#)
005      IF(X1)1,1,2
006      2      DP(A073,AT073,,,1)
007      1      IF(X2)3,3,4
008      4      DP(A124,AT051,,,1)
009      3      IF(X3)5,5,6
010      6      DP(A124,AT050,,,1)
011      5      DP(A049,AT399,AT049,,,1)

```

```
                KNOTEN  A124
001 '          X100=X100+TIME(2)
002          IF(SACK)1,1,2
003   1        DP(A112,ATJ81,,,,,1)
004   2        DP(A052,ATJ52,,,,,1)
```

Das folgende Programmbeispiel zeigt eine Rechenroutine, die in PUMA erstellt wurde. Da in dieser Sprache nur eine arithmetische Operation pro Zeile durchgeführt werden kann (vgl. Teil 4, Punkt 2.3.2.2.2.), wird das Programm länger als in FORTRAN. Diese Restriktion macht zudem eine Zerlegung von Formeln in mehrere Zeilen notwendig. Dadurch wird die Programmierung umständlich und unübersichtlich.

Die stärkste Beschränkung liegt jedoch in der Begrenzung der Variablenzahl eines Programmes auf 115, wobei jedes Element eines Feldes als Variable gezählt wird (vgl. Teil 4, Punkt 2.3.2.2.2.).

KNOTEN EG3

Eingabevariable:

```

001      I1=IT1-1
002      IG=N
003      IF(IT2-N)1,2,2
004      1  I6=IT2
005      2  BE=1-AL
006      BEQUA=BE*BE
007      BECUB=BE*BEQUA
008      ALQUA=AL*AL
009      ALCUB=AL*ALQUA
010      7  I1=I1+1
011      IF(I1-I6)3,3,4
012      G=A+B
013      H=0.5*C
014      XG(I1)=G+H
015      DIF=XG(I1)-X(I1)
016      DIFBE=BECUB*DIF
017      A=X(I1)-DIFBE
018      DIFF=2-AL
019      DI=DIFF*DIF
020      ALF=1.5*ALQUA
021      ER=B+C
022      ALDI=ALF*DI
023      B=ER-ALDI
024      CA=ALCUB*DIF
025      C=C-CA
026      GOTO 7
027      4  IF(IT2-N)5,5,8
028      8  I4=N
029      9  I4=I4+1
030      IF(I4-IT2)6,6,5
031      6  I6=I4-N
032      BA=B*I6
033      CA=0.5*C
034      CB=CA*I6
035      CC=CB*I6
036      AA=A+BA
037      XG(I4)=AA+CC
038      GOTO 9
039      5  .....

```

IT1..... Beginnperiode der Glättung
 IT2..... Endperiode der Glättung
 AL..... Glättungsparameter
 A..... Grundwert
 B..... Trendwert
 C..... Trendwert der Trendwerte
 X(I).... Beobachtungswerte
 N..... Anzahl der Beobachtungswerte

Ausgabevariable:

XG(I)... Vorhersagewerte

Bild 4:

```

*****
*
*
*SEHR GUT.
*
*SIE HABEN DIE FUNKTION DES SIGNIFIKANZTESTS
*PRINZIPIELL VERSTANDEN.
*UNTER OBIGEN ANNAHMEN MUSS T(IN) GROSSER ALS T
*(OUT) SEIN, DA DIE ZU ERREICHENDE SICHERHEITSWAHR-
*SCHEINLICHKEIT FUER DIE ABLEHNUNG DER NULLHYPOTHE-
*SE BEI DER AUFNAHME EINES NEUEN BI GROSSER SEIN
*MUSS, ALS BEI DER AUSSCHLIESSUNG EINES BI.
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*****

```

Bild 5:

```

*****
*
*
* KANN IM LAUFE DER BERECHNUNG FUER EIN NEUES BI DIE
* NULLYPOTHESE NICHT ABGELEHNT WERDEN (T KLEINER
* (TIN) ), MUSS EIN NEUES BI AUSGEWAHLT WERDEN, UND
* DIE PRUEFUNG WIEDERHOLT SICH.
* STELLT SICH HERAUS, DASS EIN BEREITS AUFGENOMMENES
* BI NICHT MEHR GESICHERT IST ( T KLEINER (TOUT) ),
* WIRD DAS BETREFFENDE BI AUS DEM REGRESSIONS-
* ERGEBNIS ELIMINIERT. DAS BEDEUTET, DASS DER
* REGRESSIONSANSATZ FUER DAS BI ZURUECKENTWICKELT
* WERDEN MUSS.
* WUENSCHEN SIE GENAEUERE INFORMATIONEN UEBER DEN
* LINDAU DES T-TESTES IN DIE SCHRITTWEISE REGRES-
* SIONSANALYSE, GEBEN SIE BITTE FORMEL EIN.
*
*
*
*
*****

```

Bild 6:

```

*1
*
* DIE BEDINGUNG DER ZIELFUNKTION LAUTET:
* DIE SUMME DER ABWEICHUNGSQUADRATE DER BEOBSACH-
* TUNGSWERTE VON DEN WERTEN DES REGRESSIONSERGE-
* BNISSES IST EIN MINIMUM.
*
* DIE NULLHYPOTHESE DES SIGNIFIKANZTESTS IST:
* DER ERWARTUNGSWERT DES  $\theta$  IST NULL.
*
* SIE ERINNERN SICH SICHER NOCH AN DIE EXPONENTIELLE
* GLAETUNG.
*
* DIE GRUNDIDEE DES EXPONENTIELLEN GLAETTENS STEHT
* IN EINER GEWICHTUNG DER BEOBSACHTUNGSWERTE NACH
* EINEM IN RICHTUNG VERGANGENHEIT ABNEHMENDEN GE-
* WICHT. WIE WERDEN BEI DER REGRESSION DIE VERGAN-
* GENHEITSWERTE GEWICHTET.
*
*      1      GLEICH      2      EXPONENTIELL
*
*
*
*

```

Bild 7:

```
*****
*
*
*SEHR GUT.
*WENN IHRE ANTWORT NICHT GERATEN IST, HABEN SIE DAS
*WESENTLICHE PRINZIP DER REGRESSIONSANALYSE VER-
*STANDEN.
*WIESSE GLEICHGEWICHTUNG ALLE! BEOBSACHTUNGSWERTE IST
*WASER GERADE FUER DIE HYPOTHETISCHEN UND OFT NICHT
*KAUSAL BEGRUENDBAREN REGRESSIONSANSATZE FUER KURZ
*FRISTIGE PROGNOSEERSTELLUNG IN DER WIRTSCHAFT
*NICHT VON VORTEIL UND MAN FUEHRTE DAHER
*FOLGENDE KORREKTUR EIN.
*
*
*
*
*
*
*
*
*
*
*
*
*****
```

Bild 8:

```
*****
*
*
*ANSTATT DIE BEOBSACHTUNGSWERTE UNABHAENGIG VOM
*ALTER IMMER GLEICH ZU GEWICHTEN UND DAMIT AUCH
*IHREN EINFLUSS AUF DAS REGRESSIONSERGEBNIS GLEICH
*ZU HALTEN, GEWICHTET MAN DIE GLIEDER DES NORMAL-
*GLEICHUNGSSYSTEMES UND DER ZIELFUNKTION MIT EINEM
*FAKTOR, DER VOM WERT 1 FUER DIE GEGENWARTSBEOBSACH-
*UNG AUSGEMOEN EXPONENTIELL BIS ZUM THEORETISCHEN
*WERT NULL (BEI UNENDLICHEN BEOBSACHTUNGSWERTEN)
*ABNIMMT, DAMIT WIRKT SICH EINE GLEICH HOHE ABWEICH-
*UNG DES BEOBSACHTUNGSWERTES VOM REGRESSIONSWERT IN
*DER VERGANGENHEIT NICHT SO STARK AUF DEN WERT DER
*ZIELFUNKTION AUS, ALS BEIM AKTUELLSTEN BEOBSACHT-
*UNGSWERT. DIE REGRESSIONSERGEBNISSE SIND GEGEN-
*WARTSBEZOGENER.
*
*
*
*
*
*****
```

Bild 9:

```
*****
*
*
*DIESE ERWEITERUNG DER SCHRITTWEISEN REGRESSION
*WURDE VON G. MATT EINGEFUEHRT.
*DER GEWICHTUNGSAKTOR HAT DIE FORM
* (1 - ALFA)**(T-V),
*WOBEI T DIE LAENGE DES BEOBSACHTUNGSZEITRAUMES
*WIEDERGIBT UND V DEN VARIABLEN LAUFINDEX, DER VON
*O (AELTESTER BEOBSACHTUNGSWERT) BIS T (JUENGSTER
*BEOBSACHTUNGSWERT) GEHT.
*ALFA IST EIN GLAETTUNGSPARAMETER, DER ZWISCHEN
*NULL UND EINS LIEGEN MUSS.
*
*
*
*
*
*
*
*
*
*
*
*****
```


Bild 13:

```
*****
*
*
*ZUSAMMENFASSEND KANN MAN FESTSTELLEN, DASS DIE
*SCHRITTWEISE REGRESSION MIT A.E.K. EINE SEHR
*EFFEKTIVE KOMBINATION VON
*  SCHRITTWEISER REGRESSIONSANALYSE
*
*  *****
*UND EXPONENTIELLER GEWICHTUNG (A.E.K.) IST. DIE
*SCHRITTWEISE REGRESSIONSANALYSE DUERFTE AUCH,
*SPEZIELL UNTER BEACHTUNG DER WESENTLICHEN AN-
*NAHMEN, DIE DEN BEHANDELTEN PROGNOSEERSTELLUNGS-
*TECHNIKEN ZUGRUNDE LIEGEN, DAS LEISTUNGSFAHIG-
*STE ALLER BEKANNTEN PROGNOSEVERFAHREN SEIN.
*BITTE ERG/ENZEN SIE OBIGE LUECKE UND VERGLEICHEN
*SIE IHRE ANTWORT MIT DEM NAECHSTEN BILD.
*
*
*
*
*
*****
```

2.2.4. Zusammenfassung

PUMA erfordert gegenüber einer frameorientierten Sprache vom Autor ein Umdenken bezüglich des Programmaufbaues. Die Gliederung in Knoten dürfte aber auch nach längerer Übung in dieser Sprache höhere Anforderungen an das logische Denken des Autors stellen, als dies beim Frame-Aufbau oder bei einer rein befehlsorientierten Sprache der Fall ist. Die hier gezeigte Liste eines PUMA-Programmes wird vom System wesentlich besser dokumentiert, wenn sich der Autor neben den Statements auch die Texte ausgeben läßt. Dies wurde hier aus Platzgründen nicht gemacht.

2.3. APL

2.3.1. Didaktogramm

Da uns die Sprache APL¹⁾ auf dem Rechner des Universitätsrechenzentrums nicht zur Verfügung stand, konnten unsere Tests nur anhand eines kleinen Programmbeispiels angestellt werden, das bei der IBM Deutschland in Stuttgart durchgeführt wurde. Dieses ist in folgendem Didaktogramm beschrieben:

● Dialogablauf

Vgl. Abb. 5.

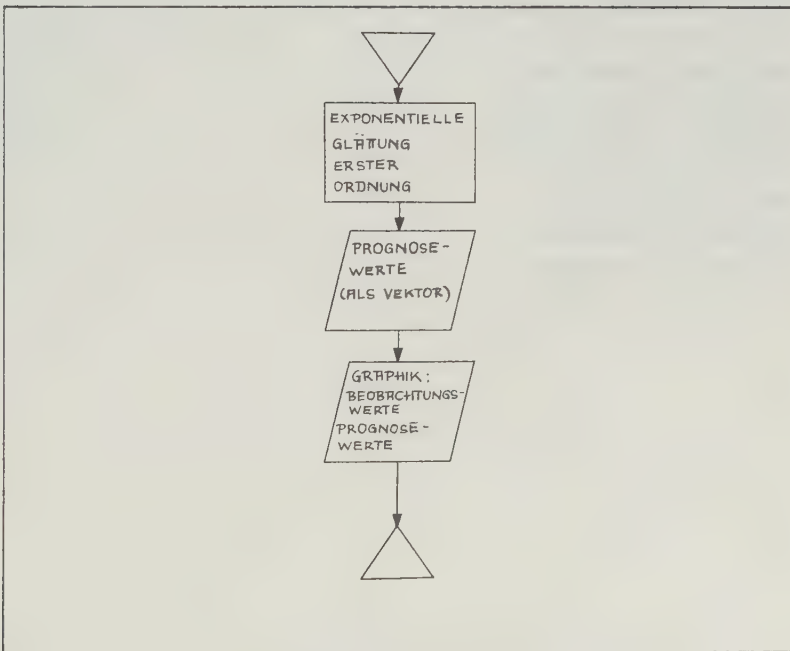


Abb. 5: Trainingsprogramm für exponentielle Glättung erster Ordnung

1) IBM (Hrsg.), APL \ 360-OS and APL \ 360-DOS, Users Manual, IBM Form GH20-0906-0.

● Verfahrensbeschreibung

Siehe Punkt 2.4.1.

● Befehle

(1) Steuerung des Ablaufes:

- EG1..... Befehl zur Durchrechnung des Verfahrens

(2) Zuweisungsbefehle:

- X(I)..... Beobachtungswerte
- IT1..... Beginnperiode der Glättung
- IT2..... Endperiode der Glättung
- A..... Grundwert
- AL..... Glättungsparameter

2.3.2. Programm

```

V EG
[1]  N=0
[2]  M=INT2
[3]  C=IT1+1
[4]  XG=M*0
[5]  XG[IT1]+(AL*X[IT1])+(1-AL)*A
[6]  XG[C]+(AL*X[C])+(1-AL)*XG[C-1]
[7]  C=C+1
[8]  +(C-M)/6
[9]  +(M-IT2)/12
[10] ' XG: ',XG+XG,(M-IT2)*C
[11] ' SO PLOT XG AND X VS XT+1N
[12] ' +C
[13] ' XT: ',XF+XG,(IT2-N)*XG[N]
[14] X=X,(IT2-N)*0
[15] ' SO PLOT XF AND X VS XT+1IT2
V
```

2.3.3. Dialogbeispiel

In diesem Beispiel befindet sich der Student unmittelbar in APL. Seine Aktivitäten bestehen in der Parameterveränderung, der Dateneingabe und dem Aufruf des Glättungsverfahrens.

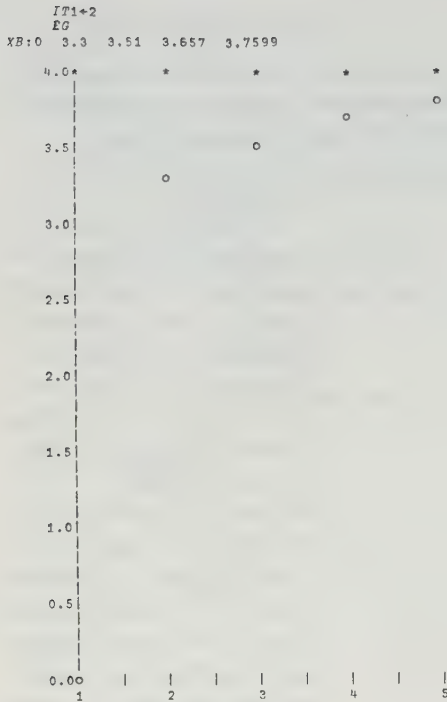
Dialog:

```
Y←4 4 4 4 4
IT←1
IT2←5
A←3
AL←0.3
FG
XR:3.3 3.51 3.657 3.7599 3.83193
```

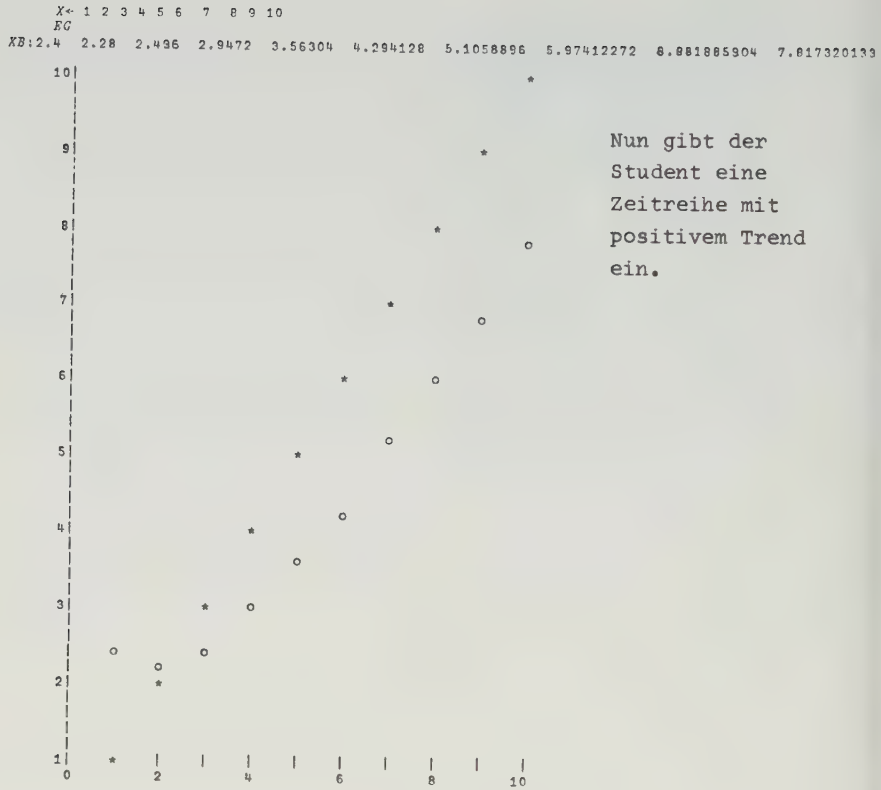
X	Y (Observation)	Y (Smoothed)
1	4.0	3.3
2	3.51	3.51
3	3.657	3.657
4	3.7599	3.7599
5	3.83193	3.83193

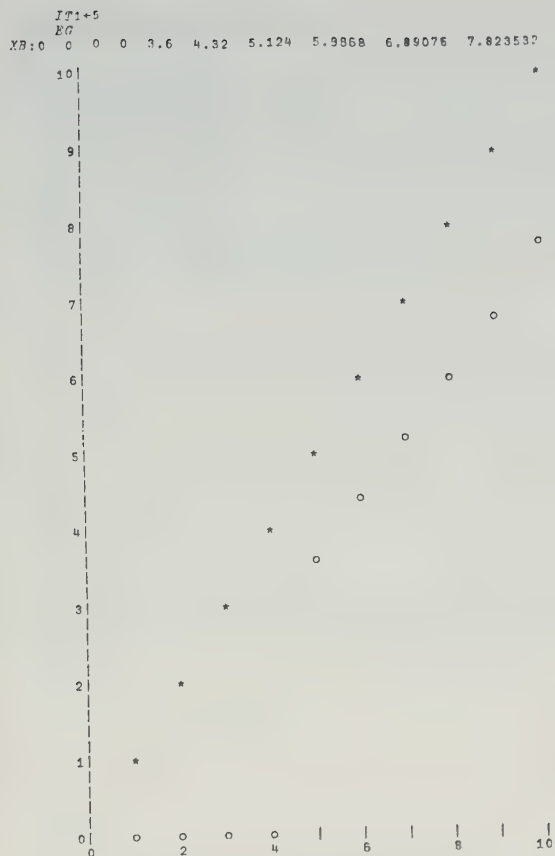
Erklärung:

Der Student gibt zuerst seine Zeitreihe ein. Dann weist er den einzelnen Parametern Werte zu und ruft das Verfahren der exponentiellen Glättung erster Ordnung mit dem Befehl EG auf. Er erhält daraufhin als Ausgabe den Vektor der Glättungswerte. Die Funktion PLOT stellt ihm den Vektor der Beobachtungswerte und der Glättungswerte graphisch dar. Die Beobachtungswerte werden als * und die Glättungswerte als o ausgegeben.



Der Student verändert die
Beginnperiode IT1 und
rechnet das Verfahren
erneut durch. Da nun kein
Glättungswert für die
Periode 1 vorliegt, die
Länge der Ausgabevektoren
aber gleich sein muß,
wird das erste Element
des Glättungsvektors
Null gesetzt. Der Maßstab
der Graphik ist daher
gegenüber der ersten Aus-
gabe verändert.





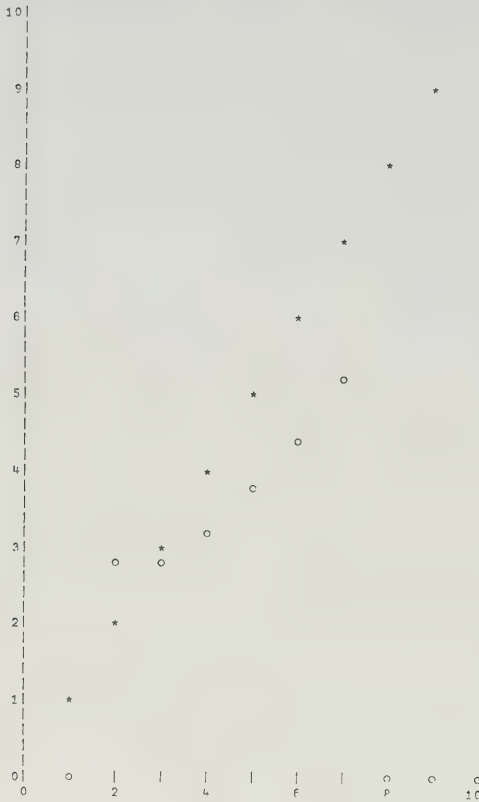
Jetzt läßt er die
Glättung bei Periode
5 beginnen.

```

IT1+2
IT2+7
EG
XC:0 2.7 2.79 3.153 3.7071 4.39497 5.176479 0 0 0

```

Schließlich läßt
er das Verfahren
von Periode 2 bis 7
laufen.



```

v
v
EG [ ]

```

□:

□:)OFF

```

000 9.46.33 09/15/72 TZS
CONNECTED 0.24.02 TO DATE 3.42.17
CPU TIME 0.01.17 TO DATE 0.03.03

```

2.3.4. Zusammenfassung

An diesem kleinen Beispiel wird die kompakte Schreibweise von APL recht deutlich. Diese hat den Vorteil einer kurzen Eingabe und kurzer Programmlisten, stört aber die Dokumentationsfähigkeit stark. Die Iverson-Notation ist nicht unbedingt die ideale Sprache für Unterrichtsprogramme, da man bei Lehrprogrammautoren keine sehr große Programmiererfahrung voraussetzen kann.

Die Antwortzeit des Systems konnte zwar bei den Versuchen nicht repräsentativ erhoben werden, dürfte aber überaus befriedigend sein. Dies geht allein schon aus der CPU-Zeit hervor, die zur Programmierung des Beispiels und zur Bearbeitung des Dialoges benötigt wurde. Sie liegt mit annähernd zwei Sekunden (auf dem System IBM 360/50) auch ökonomisch gesehen in einer akzeptablen Größenordnung.

Da wir dem Anforderungsprofil jedoch nur Systeme gegenüberstellen wollten, die wir hinreichend testen konnten, verzichten wir in Teil 4 auf eine Schilderung von APL.

2.4. TS

Im TS wurde eine Reihe von Versuchen unternommen, die klären sollten, ob und wie das Training realisiert werden kann. Wir stellen hier einen Versuch dar, der vor allem darüber Aufschluß geben sollte, wie der Dialogablauf am günstigsten gestaltet wird.

2.4.1. Didaktogramm

Da das hier gezeigte Programm in die Trainingslektion für Prognose eingebaut ist, bestehen Verbindungen zu anderen (hier nicht beschriebenen) Programmteilen¹⁾. Diese sind

1) Vgl. Reitbauer, F., a.a.O., Punkt 4.

mit gestrichelten Linien angedeutet.

● Dialogablauf

Siehe Abb. 6.

● Rechenroutinen¹⁾

Die exponentielle Glättung geht von der Annahme aus, daß ein Prozeß ξ vorhanden ist, der durch das Polynom

$$\xi(t) = \sum_{i=0}^n \frac{a_i}{i!} t^i$$

$a_1, \dots, a_n$ Glättungskoeffizient

beschrieben werden kann.

Ist ein Störfaktor vorhanden, gilt

$$x(t) = \xi(t) + \varepsilon(t).$$

$x(t)$ Beobachtungswert zum Zeitpunkt t ,

$\varepsilon(t)$ Fehlerabweichung zum Zeitpunkt t .

EG1:

Die exponentielle Glättung erster Ordnung geht daher vom Prozeß $\xi(t) = a_0$ aus und verwendet folgende Gleichung zur Bestimmung des Koeffizienten a_0 :

$$a_0(t) = x(t) + \beta(\hat{x}_1(t-1) - x(t)).$$

$$\beta = 1 - \alpha$$

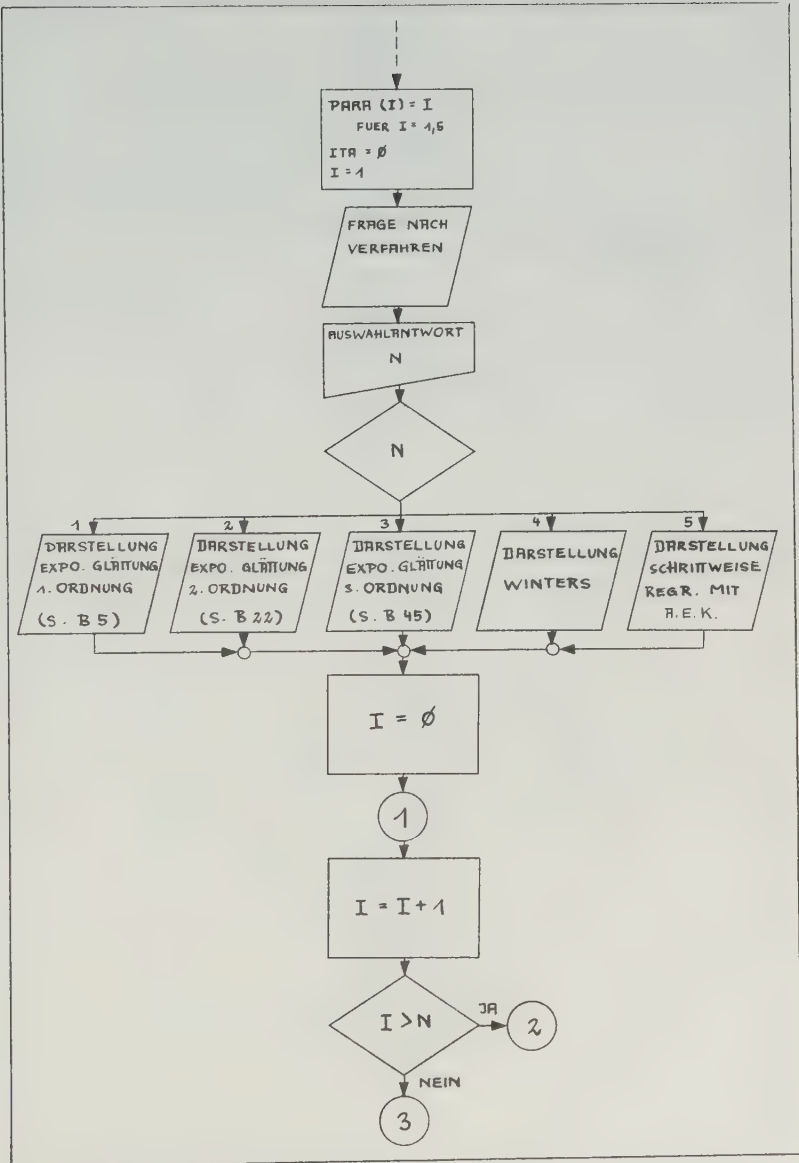
α Glättungsparameter.

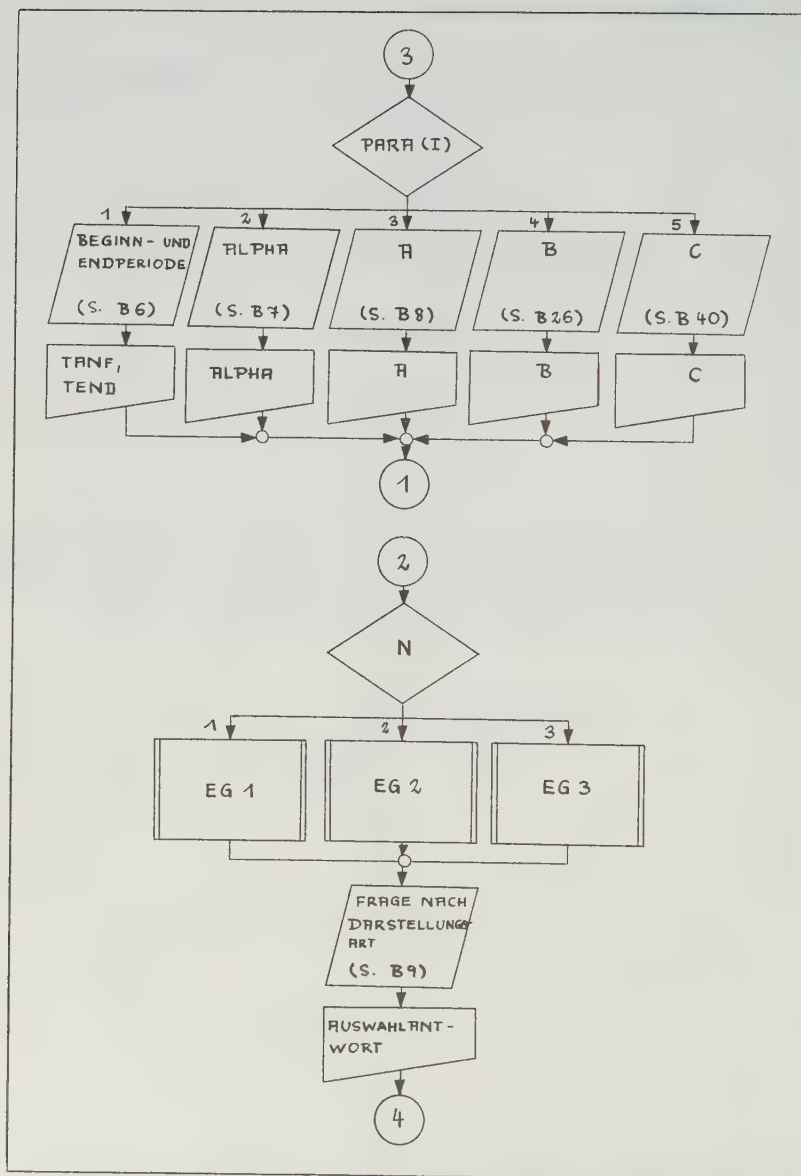
Die Vorhersage $\hat{x}$ zum Zeitpunkt t für k Perioden lautet:

$$\hat{x}_k(t) = a_0(t).$$

1) Vgl. Brown, R.G., Smoothing, Forecasting and Prediction of Discrete Time Series, Englewood Cliffs 1963, S. 143 f.

Dialogablauf:





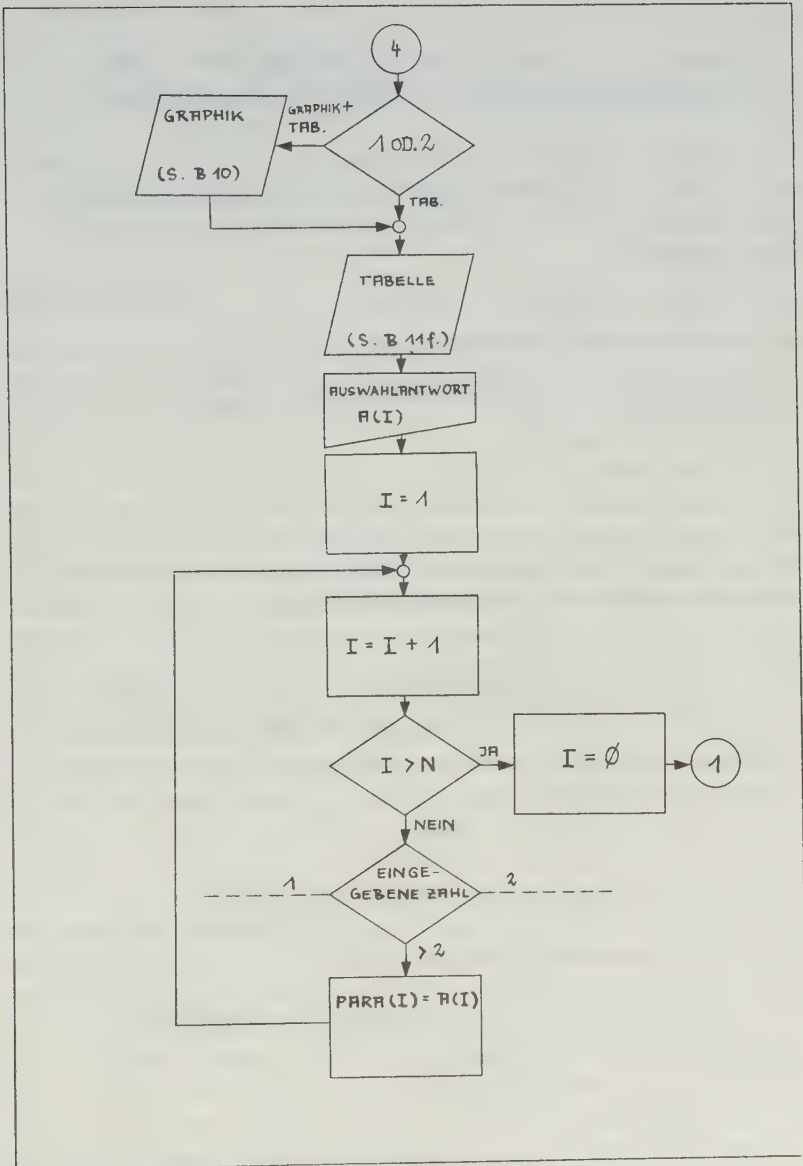


Abb. 6: Trainingsprogramm für exponentielle Glättung erster, zweiter und dritter Ordnung

EG2:

Die exponentielle Glättung zweiter Ordnung beruht auf einem linearen Modell mit folgender Prozeßbeschreibung:

$$\xi(t) = a_0 + a_1 t$$

Mit a_0 wird der Grundwert, mit a_1 der Trendwert der Zeitreihe erfaßt. Die beiden Koeffizienten werden folgendermaßen bestimmt:

$$a_0(t) = x(t) + \beta^2(\hat{x}_1(t-1) - x(t))$$

$$a_1(t) = a_1(t-1) - \lambda^2(\hat{x}_1(t-1) - x(t)).$$

Daraus ergibt sich folgende Vorhersagegleichung:

$$\hat{x}_k(t) = a_0(t) + k a_1(t).$$

EG3:

Der exponentiellen Glättung dritter Ordnung liegt ein quadratisches Modell folgender Art zugrunde:

$$\xi(t) = a_0 + a_1 t + \frac{a_2}{2} t^2$$

Mit a_2 wird der Trend der Trendwerte, also die quadratische Komponente einer Zeitreihe, in das Modell aufgenommen.

Folgende Gleichungen werden zur Abschätzung der drei Koeffizienten benutzt:

$$a_0(t) = x(t) + \beta^3(\hat{x}_1(t-1) - x(t))$$

$$a_1(t) = a_1(t-1) + a_2(t-1) - \frac{3\lambda^2}{2} (2-\lambda)(\hat{x}_1(t-1) - x(t))$$

$$a_2(t) = a_2(t-1) - \lambda^3(\hat{x}_1(t-1) - x(t))$$

Die Vorhersage wird nun auf folgende Weise berechnet:

$$\hat{x}_k(t) = a_0(t) + k a_1(t) + \frac{k^2}{2} a_2(t).$$

● Befehle

Zur Steuerung des Ablaufes:

- BEGINN.... Sprung auf den Beginnpunkt des Trainingsprogrammes
- DATEN..... Sprung in die Datenbereitstellung
- OFF..... Sprung zum Programmende

2.4.2. Programm

Aus Gründen der Übersichtlichkeit und der Einfachheit werden im TS Dialogstruktur, Rechenroutinen und Textkonserven getrennt eingegeben.

Das Unterprogramm EXPO, das den Dialog im Training der exponentiellen Glättung steuert, übernimmt aus dem Hauptprogramm des Prognosekurses den Vektor der Beobachtungswerte (Y(I)), die Mächtigkeit dieses Vektors (N) sowie den Parameter MORD, der angibt, welches Verfahren der Student gewählt hat. Es übergibt den Parameter IREA, der zur Globalsteuerung notwendig ist, an das Hauptprogramm. In LN (line) 9 bis LN 18 wird dem Studenten je nach Verfahren, das er gewählt hat, eine kurze mathematische Beschreibung ausgegeben und in einer Variablen (IALTA) vorgemerkt, welche Verfahrensparameter von ihm erfragt werden müssen (zur Textausgabe im TS s. Teil 4, Punkt 2.3.3.2.3.).

LN 26 bis LN 29 veranlassen, daß dem Studenten die Aufforderungen zur Parametereingabe in der Reihenfolge ausgegeben werden, die in LN 21 bis LN 26 festgelegt wird. Mit LN 30 bis LN 35 wird die Aufforderung zur Eingabe vorbereitet und ausgegeben. Die Überprüfung der Studenteneingabe mit den diversen Fehlermeldungen findet in LN 40 bis LN 58 statt. Analog sind auch die Eingaben der Parameter ALFA (s. Bild 7) in LN 92 bis LN 110, A (s. Bild 8) in LN 60 bis LN 70, B (s. Bild 26) in LN 72 bis LN 80, C (s. Bild 45) in LN 82 bis LN 90 aufgebaut.

Die Ausgabe der Aufforderungen, Parameterwerte einzugeben, und die Analyse der Eingaben sind im TS relativ aufwendig zu programmieren. Ungeübte Programmautoren haben bei den derzeit vorhandenen Möglichkeiten im TS Schwierigkeiten, eine für den Dialogbenutzer komfortable Ein-/Ausgabe zu erstellen (vgl. Teil 4, Punkt 2.3.4.2.3. und 2.3.4.2.4.). Nach der Parameterfestsetzung wird das vom Studenten gewählte Verfahren aufgerufen (LN 111 bis LN 117). Die Ausgliederung der Rechenroutinen aus der Dialogstruktur hat neben der bereits erwähnten größeren Übersichtlichkeit den Vorteil, daß die Routinen auch aus anderen Dialogstrukturen aufgerufen werden können (vgl. Teil 4, Punkt 2.3.2., insbesondere 2.3.2.1. und 2.3.2.2.3.). In LN 118 bis LN 125 erfolgt die vom Studenten gewünschte Ausgabe. Die Definition einer Graphik (s. Bild 10) benötigt nur zwei Zeilen (LN 123 und LN 124) und ist somit - abgesehen von einem Mangel an Dokumentationsfreundlichkeit - sehr günstig gelöst (vgl. Teil 4, Punkt 2.3.3.1.2.2.3. und 2.3.3.2.3.). Dagegen bedeutet es im TS wesentlich mehr Aufwand, die Ergebnisse mit den erforderlichen Kopfinformationen in einer Tabelle (s. Bild 11) darzustellen. Die Vorbereitung dieser Ausgabe erfolgt im Unterprogramm EOUT, das mit Modifikationen auch für andere Trainingsprogramme verwendet werden kann¹⁾ (vgl. Teil 4, Punkt 2.3.3.2.3.). In LN 130 bis LN 133 werden die Fortsetzungswünsche des Studenten (s. Bild 12) registriert und an den Verteiler in LN 29 übergeben.

An diesem Programmbeispiel sieht man, daß auch im Training Programnteile verallgemeinert werden können (vor allem bei der Ein-/Ausgabe, aber auch im Dialogablauf). Die Möglichkeit, als Programmautor Makrobefehle zu definieren, dürfte die Erstellung von Trainingsprogrammen wesentlich erleichtern (vgl. Teil 4, Punkt 2.3.1.1.).

1) EOUT wird daher beschrieben in:

Reitbauer, F., a.a.O., Teil 4.

In der Arbeit von Reitbauer, der bei der Erstellung des Trainingsprogrammes mitgewirkt hat, werden auch die anderen Unterprogramme, die hier nicht dokumentiert sind, erklärt.

USASI FORTRAN(2,1)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 03/18

```

LN 0001      SUBROUTINE EYPO(Y,N,IREA,MORD)
LN 0002      COMMON IDISP(125)
LN 0003      INTEGER EART
LN 0004      DIMENSION Y(40),PARAM(40),XG(40),KAHL(10),RA(40),PB(40),RC(40)
LN 0005      DO 215 IX=1,40
LN 0006      RA(IX)=0.
LN 0007      RB(IX)=0.
LN 0008      215 RC(IX)=0.
LN 0009      GOTO(3,1,3J2,203),MORD
LN 0010      301 IDIS=5
LN 0011      IALTA=3
LN 0012      GOTO 204
LN 0013      3J2 IDIS=60
LN 0014      IALTA=4
LN 0015      GOTO 204
LN 0016      203 IDIS=70
LN 0017      IALTA=5
LN 0018      204 CALL DISPGUT(IDIS)
LN 0019      BUFFER IN(9,J)(IDISP(1),IDISP(125))
LN 0020      IF(IFUNIT(9).NE.0)STOP3J0
LN 0021      KAHL(4)=3
LN 0022      KAHL(5)=4
LN 0023      KAHL(3)=2
LN 0024      KAHL(2)=5
LN 0025      KAHL(1)=6
LN 0026      MM=IALTA
LN 0027      114 DOJ1L=1,MM
LN 0028      L1 = KAHL(L)
LN 0029      GOTO(103,116,207,2J8,117,115),L1
LN 0030      115 IDIS=27
LN 0031      N2=2
LN 0032      PARAM(1)=IT1
LN 0033      PARAM(2)=IT2
LN 0034      K=1
LN 0035      CALL TAB(IDIS,N2,PARAM,K,0,16,17,1,4,4,1)
LN 0036      CALL DISPOUT(G)
LN 0037      104 EART=2
LN 0038      NN=2
LN 0039      CALL DISPIN(EART,PARAM,NN,IREA,IDIS,IALTA)
LN 0040      IF(IREA-1)102,103,103
LN 0041      102 IF(PARAM(1).LE.PARAM(2))GOTO 16J
LN 0042      IDIS=63
LN 0043      CALL DISPOUT(IDIS)
LN 0044      GOTO 164
LN 0045      16J IF(PARAM(2).LE.40)GOTO 105
LN 0046      IDIS=65
LN 0047      CALL DISPOUT(IDIS)
LN 0048      GOTO104
LN 0049      105 IF(PARAM(1).LE.FLOAT(N))GOTO 201
LN 0050      IDIS=68
LN 0051      CALL DISPOUT(IDIS)
LN 0052      GOTO104
LN 0053      201 IF(PARAM(1).GT.0)GOTO202
LN 0054      IDIS=64

```

USASI FORTRAN(2,1)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 03/1

```

LN 0055      CALL DISPOUT(IDIS)
LN 0056      GOTO104
LN 0057      202  IT1=PARAM(1)
LN 0058      IT2=PARAM(2)
LN 0059      GOTO101
LN 0060      116  IOIS=30
LN 0061      N2=1
LN 0062      PARAM(1)=AA
LN 0063      K=1
LN 0064      CALL TAB(IOIS,N2,PARAM,K,2,16,16,1,4,4,1)
LN 0065      CALL DISPOUT(I)
LN 0066      EART=2
LN 0067      NN=1
LN 0068      CALL DISPIN(EART,PARAM,NN,IREA,IOIS,IALTA)
LN 0069      IF (IREA-1)106,103,103
LN 0070      106  AA=PARAM(1)
LN 0071      GOTO101
LN 0072      207  IOIS=71
LN 0073      N2=1
LN 0074      PARAM(1)=BB
LN 0075      K=1
LN 0076      CALL TAB(IOIS,N2,PARAM,K,3,16,16,1,4,4,1)
LN 0077      CALL DISPOUT(I)
LN 0078      CALL DISPIN(2,PARAM,1,IREA,71,IALTA)
LN 0079      IF (IREA-1)209,103,103
LN 0080      209  BB=PARAM(1)
LN 0081      GOTO 101
LN 0082      208  IOIS=61
LN 0083      N2=1
LN 0084      PARAM(1)=CC
LN 0085      K=1
LN 0086      CALL TAB(IOIS,N2,PARAM,K,3,16,16,1,4,4,1)
LN 0087      CALL DISPOUT(I)
LN 0088      CALL DISPIN(2,PARAM,1,IREA,61,IALTA)
LN 0089      IF (IREA-1)210,103,103
LN 0090      210  CC=PARAM(1)
LN 0091      GOTO 101
LN 0092      117  IOIS=31
LN 0093      N2=1
LN 0094      PARAM(1)=AL
LN 0095      K=1
LN 0096      CALL TAB(IOIS,N2,PARAM,K,5,16,16,1,4,4,1)
LN 0097      CALL DISPOUT(I)
LN 0098      109  EART=2
LN 0099      NN=1
LN 0100      CALL DISPIN(EART,PARAM,NN,IREA,IOIS,IALTA)
LN 0101      IF (IREA-1)107,103,103
LN 0102      107  IF (PARAM(1).LE.1.)GOTO108
LN 0103      IOIS=63
LN 0104      CALL DISPOUT(IDIS)
LN 0105      GOTO 103
LN 0106      108  IF (PARAM(1).GE.3.)GOTO110
LN 0107      IOIS=63
LN 0108      CALL DISPOUT(IDIS)

```

USASI FORTRAN(2,1)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 03/18

```

LN 0103      GOTO 109
LN 0110      113 AL=PARAM(1)
LN 0111      101 CONTINUE
LN 0112      GOTO(211,212,213),MORD
LN 0113      211 CALL EG1(N,Y,XG,AA,RA,AL,IT1,IT2)
LN 0114      GOTO 214
LN 0115      212 CALL EG2(N,Y,XG,AA,BB,RA,R3,AL,IT1,IT2)
LN 0116      GOTO 214
LN 0117      213 CALL EG3(N,Y,XG,AA,BB,CC,RA,R3,RC,AL,IT1,IT2)
LN 0118      214 CALL DISPOUT(72)
LN 0119      CALL DISPIN(1,PARAM,2,IREA,72,IALTA)
LN 0120      IF(IREA-1)216,103,103
LN 0121      216 IA=PARAM(1)
LN 0122      GOTO(218,217),IA
LN 0123      217 M2=IT2-IT1+1
LN 0124      CALL PLOT(2,Y,1,N,1,XG,IT1,M2,1,Y,1,N,0)
LN 0125      218 CALL EOUT(N,IT1,IT2,Y,XG,RA,RD,RC,AL,MORD)
LN 0126      EART=1
LN 0127      NN=6
LN 0128      CALL DISPIN(EART,PARAM,NN,IREA,DIS,IALTA)
LN 0129      IF(IREA-1)112,103,103
LN 0130      112 MM=IALTA
LN 0131      DO113L=1,MM
LN 0132      113 KAHL(L)=PARAM(L)
LN 0133      GOTO 114
LN 0134      103 RETURN
LN 0135      END

```

USASI FORTRAN DIAGNOSTIC RESULTS FOR EXPO

NO ERRORS

USASI FORTRAN(2,1)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 03/18

```

LN 0001      SUBROUTINE EG1(N,Y,XG,AA,RA,AL,IT1,IT2)
LN 0002      DIMENSION Y(40),XG(40),RA(40)
LN 0003      A=AA
LN 0004      IF(IT2.LE.N)GOTO6
LN 0005      I6=N
LN 0006      RA(IT1)=A
LN 0007      GOTO7
LN 0008      6 I6=IT2
LN 0009      7 DO 111=IT1,I6
LN 0010      XG(11)=A
LN 0011      A=AL*Y(11)+(1.-AL)*
LN 0012      1 RA(11+1)=A
LN 0013      IF(IT2-I6)4,4,8
LN 0014      3 I4=I6+1
LN 0015      DO 515=I4,IT2
LN 0016      5 XG(515)=A
LN 0017      4 CONTINUE
LN 0018      RETURN
LN 0019      END

```

USASI FORTRAN DIAGNOSTIC RESULTS FOR EG1

NO ERRORS

USASI FORTRAN(2.1)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 03/1

```

LN 0001      SUBROUTINE EG2(N,X,XG,AA,BB,RA,RB,AL,IT1,IT2)
LN 0002      DIMENSION X(4),XG(4),RA(4),RB(4)
LN 0003      A=AA
LN 0004      B=BB
LN 0005      I3=N
LN 0006      I6=I3
LN 0007      IF(IT2-I3)6,7,7
LN 0008      6 I6=IT2
LN 0009      7 BE=1.-AL
LN 0010      BEQUA=BE**2
LN 0011      ALQUA=AL**2
LN 0012      RA(IT1)=A
LN 0013      RB(IT1)=B
LN 0014      DO 1 I1=IT1,I6
LN 0015      XG(I1)=A*B
LN 0016      OIF=XG(I1)-X(I1)
LN 0017      A=X(I1)+BEQUA*OIF
LN 0018      B=9-ALQUA*OIF
LN 0019      RB(I1+1)=B
LN 0020      1 RA(I1+1)=A
LN 0021      IF(IT2-I3)4,4,3
LN 0022      3 I4=I3+1
LN 0023      DO 5 I5=I4,IT2
LN 0024      I6=I5-I3
LN 0025      5 XG(I5)=A+B*I6
LN 0026      4 CONTINUE
LN 0027      RETURN
LN 0028      END

```

USASI FORTRAN DIAGNOSTIC RESULTS FOR EG2

NO ERRORS

USASI FORTRAN(2.1)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 03/1

```

LN 0001      SUBROUTINE EG3(N,X,XG,AA,BB,CC,RA,RB,RC,AL,IT1,IT2)
LN 0002      DIMENSION X(4),XG(4),RA(4),RB(4),RC(4)
LN 0003      A=AA
LN 0004      B=BB
LN 0005      C=CC
LN 0006      I3=N
LN 0007      I6=I3
LN 0008      IF(IT2-I3)6,7,7
LN 0009      6 I6=IT2
LN 0010      7 BE=1.-AL
LN 0011      BECUB=BE**3
LN 0012      ALCUB=AL**3
LN 0013      RA(IT1)=A
LN 0014      RB(IT1)=B
LN 0015      RC(IT1)=C
LN 0016      DO 1 I1=IT1,I6
LN 0017      XG(I1)=A+B*I1+5*C
LN 0018      OIF=XG(I1)-X(I1)
LN 0019      A=X(I1)-BECUB*OIF
LN 0020      B=3*C-1.5*AL*AL*(2-AL)*OIF
LN 0021      C=C-ALCUB*OIF
LN 0022      RB(I1+1)=B
LN 0023      RC(I1+1)=C
LN 0024      1 RA(I1+1)=A
LN 0025      IF(IT2-I3)4,4,3
LN 0026      3 I4=I3+1
LN 0027      DO 5 I5=I4,IT2
LN 0028      I6=I5-I3
LN 0029      5 XG(I5)=A+B*I6+3.5*C*I6*I6
LN 0030      4 CONTINUE
LN 0031      RETURN
LN 0032      END

```

USASI FORTRAN DIAGNOSTIC RESULTS FOR EG3

NO ERRORS

Bild 6:

```
*****
*1,1;
*
*
*
*BITTE GEBEN SIE DIE GEWUNSCHTEN ANFANGS- UND
*ENDPERIODE, FÜR DIE GEGLETTET WERDEN SOLL, IN
*DER GENANNTEN REIHENFOLGE ALS GANZE ZAHLEN EIN,
*DIE NICHT GRÖßER ALS 43 SEIN SÖLLEN.
*
*
*
*
*BISHERIGE WERTE:
*
*   BEGINNPERIODE IT1 =      0.
*   ENDPERIODE IT2  =      4.
*
*
*EINGABE: IT1, IT2
*****
```

Bild 7:

```
*****
*.3;
*
*
*BITTE GEBEN SIE DEN GLETTUNGSFAKTOR ALFA
*ALS WERT ZWISCHEN 0 UND 1 EIN.
*
*
*
*
*BISHERIGE WERTE:
*
*   GLETTUNGSWERT ALFA = .00000
*
*
*EINGABE: GLETTUNGSWERT ALFA
*****
```

Da der Student bisher noch keinen Parameter eingegeben hat, scheint für ALFA Null auf.

Bild 8:

```
*****
*52;
*
*
*
*BITTE GEBEN SIE IHRE ABSCHÄTZUNG FÜR DEN
*GRUNDWERT BIS ZUM BEGINN DER GLETTUNG EIN.
*
*
*
*
*BISHERIGER WERT:
*
*   GRUNDWERT A =      .00
*
*
*EINGABE: GRUNDWERT
*****
```


Bild 12:

```
*****
*5,6: EXPONENTIELLE GLAETTUNG ERSTER ORDNUNG:
*VON PER. 1.JJ BIS PER. 10.00 ALFA = .30
*M.A.ABW= 3.48
*
*PERIODE BEGRIFFT.GRUNDWT.PROGNOSE ABS.ABW.*
* 13. 86.20
* 14. 96.30
* 15. 100.20
* 16. 116.40
* 17. 133.80
* 18. 152.10
* 19. 174.20
* 21. 195.20
* 22. 223.40
* 22. 256.00
*WIE SOLL ES WEITERGEHEN?
*1 BEENDEN 5 GLAETTUNGSFAKTOR AENDERN
*2 GRUNDWERT AENDERN 6 GLAETTUNGSZEITRAUM AENDERN
*
*EINGABE: AUSWAHLANTWORT ODER SEND
*****
```

Bild 13:

```
*****
*1;
*
*
*
*BITTE GEBEN SIE DEN GLAETTUNGSFAKTOR ALFA
*ALS WERT ZWISCHEN 0 UND 1 EIN.
*
*
*
*
*BISHERIGE WERTE:
*
* GLAETTUNGSWERT ALFA = .30000
*
*
*EINGABE: GLAETTUNGSWERT ALFA
*****
```

Bild 14:

```
*****
*1,27;
*
*
*
*BITTE GEBEN SIE DIE GEWUNSCHTEN ANFANGS- UND
*ENDPERIODE, FUER DIE GEGLAETTET WERDEN SOLL, IN
*DER GENANNTE REIHENFOLGE ALS GANZE ZAHLEN EIN,
*DIE NICHT GROSSEER ALS 40 SEIN SOLLEN.
*
*
*
*
*BISHERIGE WERTE:
*
* BEGINNPERIODE IT1 = 1.
* ENDPERIODE IT2 = 10.
*
*
*EINGABE: IT1, IT2
*****
```


Bild 18:

```

*****
*      EXPONENTIELLE GLAETTUNG ERSTER ORDNUNG      *
*VON PER. 1.00 BIS PER. 27.00 ALFA = 1.00 *
*M.A.ABW= 9.36 *
*
*PERIODE      BEOBACHT.GRUNDWT.PROGNOSE ABS.ABW.*
*  4.         52.30  51.30  51.30  0.33 *
*  5.         53.40  52.30  52.30  1.09 *
*  6.         55.40  53.40  53.40  2.00 *
*  7.         57.70  55.40  55.40  2.29 *
*  8.         59.90  57.70  57.70  2.19 *
*  9.         64.20  59.90  59.90  4.29 *
* 10.         68.70  64.20  64.20  4.50 *
* 11.         74.50  68.70  68.70  5.80 *
* 12.         78.80  74.50  74.50  4.30 *
* 13.         86.20  78.80  78.80  7.39 *
* 14.         96.30  86.20  86.20 10.10 *
* 15.        100.20  96.30  96.30  3.89 *
* 16.        116.40 100.20 100.20 16.20 *
* 17.        133.80 116.40 116.40 17.39 *
*EINGABE: AUSWAHLANTWORT ODER SEND
*****

```

Bild 19:

```

*****
*1: EXPONENTIELLE GLAETTUNG ERSTER ORDNUNG*
*VON PER. 1.00 BIS PER. 27.00 ALFA = 1.00 *
*M.A.ABW= 9.36 *
*
*PERIODE      BEOBACHT.GRUNDWT.PROGNOSE ABS.ABW.*
* 18.         152.10 133.80 133.80 18.29 *
* 19.         174.20 152.10 152.10 22.09 *
* 20.         135.20 174.20 174.20 20.99 *
* 21.         223.40 135.20 135.20 28.19 *
* 22.         256.00 223.40 223.40 32.60 *
* 23.          256.00 256.00 *
* 24.          256.00 *
* 25.          256.00 *
* 26.          256.00 *
* 27.          256.00 *
*WIE SOLL ES WEITERGEHEN:
*1 BEENDEN          5 GLAETTUNGSFAKTOR AENDERN
*2 GRUNDWERT AENDERN 6 GLAETTUNGSZEITRAUM AENDERN*
*
*EINGABE: AUSWAHLANTWORT ODER SEND
*****

```

Bild 20:

```

*****
*1:
*
*HOLLEN SIE EINES DER ANDEREN VORHANDENEN
*PROGNOSEVERFAHREN EINSETZEN:
*
* 1 JA
*
* 2 NEIN
*
*
*EINGABE: AUSWAHLANTWORT.
*****

```

Bild 21:

```
*****
*2;
*
*SIE KOENNEN DIE PARAMETER IHRES
*PROGNOSEMODELLES MIT HILFE
*FOLGENDER VERFAHREN BERECHNEN:
*
*      1 EXPONENTIELLES GLAETTEN 1. ORDNUNG
*      2 EXPONENTIELLES GLAETTEN 2. ORDNUNG
*      3 EXPONENTIELLES GLAETTEN 3. ORDNUNG
*      4 WINTERS-VERFAHREN
*      5 SCHRITTWEISE REGRESSION MIT A.E.K.
*
*
*
*
*EINGABE: AUSWAHLANTWORT
*****
```

Bild 22:

```
*****
*
*
*BEIM LINEAREN MODELL MIT EXPONENTIELLER GLAETTUNG
*HABEN SIE ES MIT FOLGENDEM MATHEMATISCHEN MO-
*DELL ZU TUN:
*
* $SX(T+K) = A + B * K$ 
* $A = X(T) + (1-AL) * 2 * (SX(T) - X(T))$ 
* $B = B - AL * 2 * (SX(T) - X(T))$ 
*
* $SX(T+K)$ ...SCHAETZUNG DER VARIABLEN X FUEER DEN
*      ZEITPUNKT T+K
* $X(T)$ .....OBSACHTUNGSWERT DER VARIABLEN X ZUM
*      ZEITPUNKT T
*A.....GRUNDWERT
*B.....TRENDWERT
*AL.....GLAETTUNGSPARAMETER
*
*
*
*****
```

Bild 23:

```
*****
*1,LC;
*
*
*BITTE GEBEN SIE DIE GEWUNSCHTEN ANFANGS- UND
*ENDPERIODE, FUEER DIE GEGLAETTET WERDEN SOLL, IN
*DER GENANNTEN REIHENFOLGE ALS GANZE ZAHLEN EIN,
*DIE NICHT GROESSER ALS 40 SEIN SOLLN.
*
*
*
*
*BISHERIGE WERTE:
*
*      BEGINNPERIODE IT1 =      1.
*      ENDPERIODE IT2 =      27.
*
*
*EINGABE: IT1, IT2
*****
```

Bild 24:

```
*****
*3;
*
*
*BITTE GEBEN SIE DEN GLAETTUNGSFAKTOR ALFA
*ALS WERT ZWISCHEN 0 UND 1 EIN.
*
*
*
*
*BISHERIGE WERTE:
*
*   GLAETTUNGSWERT ALFA = 1.00000
*
*
*EINGABE: GLAETTUNGSWERT ALFA.
*****
```

Bild 25:

```
*****
*50;
*
*
*BITTE GEBEN SIE IHRE ABSCHAEZUNG FUEER DEN
*GRUNDWERT BIS ZUM BEGINN DER GLAETTUNG EIN.
*
*
*
*
*BISHERIGER WERT:
*
*   GRUNDWERT A =    52.00
*
*
*EINGABE: GRUNDWERT
*****
```

Bild 26:

```
*****
*2;
*
*GEBEN SIE BITTE DEN TRENDWERT ZUM BEGINN DES
*GLAETTUNGSZEITRAUMES AN.
*
*
*
*
*BISHERIGER WERT:
*
*   LINEARER TRENDWERT B =    .000
*
*
*EINGABE: TRENDWERT B
*****
```


Bild 36:

```

*****
*      EXPONENTIELLE GLAETTUNG ZWEITER ORDNUNG      *
*VON PER.      1.00BIS PER.      30.00  ALFA =      1.00  *
*M.A.ABW=      2.37  *
*
*PERIODE  BEOBSCHT. GRUNDWT. TRENDWT. PROGNOSE  ABS. ABW. *
*   7.   57.70   55.40   2.30   57.40   .29  *
*   8.   59.30   57.70   2.29   61.00   .19  *
*   9.   64.20   59.30   2.20   62.10   2.09  *
*  10.   68.70   64.20   4.23   63.43   .20  *
*  11.   74.50   68.70   4.49   73.20   1.30  *
*  12.   78.80   74.50   5.79   80.30   1.50  *
*  13.   86.20   78.80   4.23   83.10   3.10  *
*  14.   96.30   86.20   7.39   93.53   2.70  *
*  15.  100.20   96.30  10.10  106.40   6.20  *
*  16.  116.40  100.20   3.89  104.09  12.30  *
*  17.  133.30  116.40  16.20  132.60   1.19  *
*  18.  152.10  133.30  17.39  151.20   .30  *
*  19.  174.20  152.10  18.30  170.40   3.79  *
*  20.  195.20  174.20  22.09  196.29   1.09  *
*EINGABE: AUSWAHLANTWORT ODER SEND
*****

```

Bild 37:

```

*****
*1;      EXPONENTIELLE GLAETTUNG ZWEITER ORDNUNG      *
*VON PER.      1.00BIS PER.      30.00  ALFA =      1.00  *
*M.A.ABW=      2.37  *
*
*PERIODE  BEOBSCHT. GRUNDWT. TRENDWT. PROGNOSE  ABS. ABW. *
*   21.  223.40  195.20  21.00  216.20   7.19  *
*   22.  256.00  223.40  28.20  251.60   4.39  *
*   23.           256.00  32.59  288.60          *
*   24.           321.20          *
*   25.           353.80          *
*   26.           386.40          *
*   27.           418.39          *
*   28.           451.59          *
*   29.           484.19          *
*   30.           516.30          *
*WIE SOLL ES WEITERGEHEN:
*1 BEENDEN:          5 GLAETTUNGSFAKTOR AENDERN*
*2 GRUNDWERT AENDERN 6 GLAETTUNGSZEITR. AENDERN*
*3 TRENDWERT AENDERN
*EINGABE: AUSWAHLANTWORT ODER SEND
*****

```

Bild 38:

```

*****
*1;
*
*WOLLEN SIE EINES DER ANDEREN VORHANDENEN
*PROGNOSEVERFAHREN EINSETZEN:
*
*   1  JA
*   2  NEIN
*
*
*
*
*
*
*
*
*
*
*EINGABE: AUSWAHLANTWORT.
*****

```


Bild 48:

```
*****
*          EXPONENTIELLE GLAETTUNG DRITTER ORDNUNG          *
* VON PER. 1.033IS PER. 10.00ALFA = .40                    *
* M.A.ABW= .57                                              *
* * * * *
* PERIODE BEOBACHT.GRUNDWT.TRENDW1.TRENDW2.PROGNOSE *
* 1. 51.00 50.00 .10 .30 50.25 *
* 2. 51.30 51.16 .68 .34 52.02 *
* 3. 51.90 51.14 .75 .30 52.05 *
* 4. 52.30 51.86 1.00 .29 53.01 *
* 5. 53.40 52.14 1.01 .24 53.28 *
* 6. 55.40 53.42 1.30 .25 54.85 *
* 7. 57.70 55.51 1.76 .23 57.43 *
* 8. 59.30 57.75 2.16 .30 60.07 *
* 9. 64.20 59.86 2.40 .29 62.41 *
* 10. 68.70 64.58 3.38 .40 68.17 *
* 11. 74.50 68.31 3.39 .44 *
* 12. 78.00 *
* * * * *
* EINGABE: AUSWAHLANTWORT ODER SEND *
*****
```

Bild 49:

```
*****
*          EXPONENTIELLE GLAETTUNG DRITTER ORDNUNG          *
* VON PER. 1.033IS PER. 10.00ALFA = .40                    *
* M.A.ABW= .57                                              *
* * * * *
* PERIODE BEOBACHT.GRUNDWT.TRENDW1.TRENDW2.PROGNOSE *
* 13. 86.20 *
* 14. 96.30 *
* 15. 100.20 *
* 16. 116.40 *
* 17. 133.30 *
* 18. 152.10 *
* 19. 174.20 *
* 20. 195.20 *
* 21. 223.40 *
* 22. 256.00 *
* WIE SOLL ES WEITERGEHEN? *
* 1 BEENDEN 4 TRENDWERT2 AENDERN *
* 2 GRUNDWERT AENDERN 5 GLAETTUNGSWERT AENDERN *
* 3 TRENDWERT1 AENDERN 6 GLAETTUNGSZTR.AENDERN *
* EINGABE: AUSWAHLANTWORT ODER SEND *
*****
```

Bild 50:

```
*****
* 1,25; *
* * * * *
* BITTE GEBEN SIE DIE GEWUENSCHTE ANFANGS- UND *
* ENDPERIODE, FUER DIE GEGLAETTET WERDEN SOLL, IN *
* DER GENANNTEN REIHENFOLGE ALS GANZE ZAHLEN EIN, *
* DIE NICHT GROESSER ALS 40 SEIN SOLLN. *
* * * * *
* BISHERIGE WERTE: *
* BEGINNPERIODE IT1 = 1. *
* ENOBERIODE IT2 = 10. *
* * * * *
* EINGABE: IT1, IT2 *
*****
```


Bild 54:

```

*****
* EXPONENTIELLE GLAETTUNG DRITTER ORDNUNG *
* VON PER. 1.00 BIS PER. 25.00 ALFA = .40 *
* M.A.ABW= 2.34 *
*
* PERIODE BEOBACHT.GRUNDWT.TRENDW1.TRENDW2.PROGNOSE *
* 2. 51.30 51.16 .68 .34 52.02 *
* 3. 51.30 51.14 .75 .30 52.45 *
* 4. 52.30 51.86 1.00 .29 53.01 *
* 5. 53.40 52.14 1.01 .24 53.23 *
* 6. 55.40 53.42 1.30 .25 54.85 *
* 7. 57.70 55.51 1.76 .28 57.43 *
* 8. 59.90 57.75 2.16 .30 60.07 *
* 9. 64.20 59.36 2.40 .23 62.41 *
* 10. 68.70 64.59 3.38 .40 68.17 *
* 11. 74.50 68.81 3.99 .44 73.12 *
* 12. 79.80 74.81 5.00 .53 80.08 *
* 13. 86.20 74.52 5.04 .45 83.79 *
* 14. 96.30 86.72 6.42 .60 93.44 *
* 15. 100.20 96.91 8.12 .79 105.43 *
* EINGABE: AUSWAHLANTHORT ODER SEND
*****

```

Bild 55:

```

*****
* 1; EXPONENTIELLE GLAETTUNG DRITTER ORDNUNG *
* VON PER. 1.00 BIS PER. 25.00 ALFA = .40 *
* M.A.ABW= 2.34 *
*
* PERIODE BEOBACHT.GRUNDWT.TRENDW1.TRENDW2.PROGNOSE *
* 16. 116.40 99.16 6.90 .45 106.20 *
* 17. 133.90 118.60 11.27 1.10 130.43 *
* 18. 152.10 134.52 13.67 1.32 148.86 *
* 19. 174.20 152.79 16.24 1.53 169.80 *
* 20. 195.20 175.14 19.46 1.81 195.51 *
* 21. 223.40 195.13 21.15 1.79 217.17 *
* 22. 256.00 224.74 25.33 2.18 251.17 *
* 23. 257.04 23.37 2.43 237.66 *
* 24. 320.79 *
* 25. 356.41 *
* WIE SOLL ES WEITERGEHEN: *
* 1 BEENDEN 4 TRENDWERT2 AENDERN *
* 2 GRUNDWERT AENDERN 5 GLAETTUNGSWERT AENDERN *
* 3 TRENDWERT1 AENDERN 6 GLAETTUNGSZTR.AENDERN *
* EINGABE: AUSWAHLANTHORT ODER SEND
*****

```

2.4.4. Zusammenfassung

Die Trennung der Dialogprogramme in Texte, Rechenroutinen und Dialogstruktur bringt einen wesentlichen Gewinn an Übersichtlichkeit. Es wäre jedoch wünschenswert, neben dem Strukturprogramm, wie es gezeigt wurde, auch noch eine Kombination mit den Texten ausgeben zu können. Die Struktur in der dargestellten Form kann auch von einem geübten Programmautor keineswegs flüssig gelesen werden. Besonders störend wirken Unterprogrammaufrufe mit langen Parameterlisten. Die Dokumentation kann allerdings durch Einfügen von Kommentarkarten wesentlich verbessert werden.

Bei der Programmierung der Dialogstruktur macht sich vor allem das Fehlen von einigen Makrobefehlen wie beispielsweise zur Analyse der Auswahlfrage störend bemerkbar. Die Programmierung der Rechenroutinen dagegen ist durch die Verwendung von FORTRAN gut gelöst.

Die Antwortzeit des TS liegt meist unter einer Sekunde. Es ist zwar zu bedenken, daß diese Aussage nur für den Betrieb mit einem einzigen Terminal untersucht wurde, doch dürfte dieser Wert auch bei Verwendung mehrerer Datenendgeräte kaum die Sekundengrenze überschreiten. Dies liegt an der vollständig kompilierenden Verarbeitungsweise, die durch die Verwendung von FORTRAN gegeben ist. Das TS ist somit auch unter Kostengesichtspunkten zur Realisierung der Trainingsstrategie gut geeignet.

V i e r t e r T e i l

Anforderungsprofil und Realisierungsmöglichkeiten

In diesem Teil wird ein Anforderungsprofil an ein System beschrieben, das einerseits eine geeignete Sprache zur Erstellung von CET-Kursen bereitstellt und andererseits die Verwaltung des Dialogbetriebes übernimmt.

Zur Definition des Anforderungsprofiles standen uns

- die eigene Erfahrung aus den Experimenten mit den verschiedenen Systemen und
- aus den Studentenversuchen mit dem CET-Kurs über Prognoserechnung sowie
- die fremde Erfahrung, die wir aus der aktuellen Literatur über ähnliche Projekte entnahmen,

zur Verfügung.

Da sich die Versuche mit CET hauptsächlich auf Prognoserechnung bezogen haben, gelten die im folgenden beschriebenen Anforderungen streng genommen nur für dieses Wissensgebiet. Erste Versuche auf anderen Gebieten (Produktions- und Investitionsplanung sowie Break-even-Analyse) haben aber gezeigt, daß dort keine wesentlich anderen Forderungen an ein CET-System zu stellen sind.

Diese Feststellung kann auch für den Problemlösungsdialog getroffen werden, der sich dadurch vom üblichen Trainingsdialog unterscheidet, daß der Student nicht ein bestimmtes

Verfahren an mehreren Datenkonstellationen getestet, sondern eine bestimmte Problemstellung durch den Einsatz verschiedener Verfahren zu lösen versucht. Die Anforderungen an ein CET-System sind im Problemlösungsdialog primär dadurch verschieden, daß der Dialog stärker als beim Verfahrenstraining durch Befehle gesteuert wird.

Es ist hier nicht möglich, jede Anforderung bis ins letzte Detail auszuführen. Viele Funktionen, wie z.B. der Schutz der Schülerdaten und der Lernprogramme oder das Verbot von Schülerbefehlen in bestimmten Situationen, sind zu so selbstverständlichen Teilen von RUU-Systemen geworden, daß sie nicht eigens erwähnt werden müssen. Forderungen, die besonders bei der Realisierung des Entscheidungstrainings auftreten, werden weiter ausgeführt als solche, die bereits in bestehenden Systemen verwirklicht sind.

Es wird hier keine Trennung der Anforderungen unter dem Gesichtspunkt Schüler oder Lehrer vorgenommen, wie dies bei Diskussionen über RUU-Systeme oft der Fall ist. Da beide mit demselben System arbeiten müssen, soll dieses für beide möglichst gute Lösungen bieten.

1. Anforderungen an die Hardware

Das CET ist ein Lehrverfahren zur Ausbildung relativ kleiner Adressatengruppen. Während für PLATO IV¹⁾ eigene Hardwareteile geschaffen wurden, um bis zu 4000 Studenten an einen Rechner anzuschließen, muß beim CET prinzipiell von der bei potentiellen Anwendern vorhandenen Hardware ausgegangen werden. Diese muß jedoch gewisse Mindestanforderungen erfüllen, um CET zu erlauben. Für eine Minimierung der variablen Kosten wäre zwar die Erfüllung einer Reihe von

1) Bitzer, D.L., Johnson, R.L., PLATO: A Computer-Based System Used in the Engineering of Education, Control Data Institut, Frankfurt am Main o.J., S. 9 ff.

weiteren Forderungen sinnvoll, doch dürften oft die hohen Fixkosten zur Beschaffung der entsprechenden Geräte dagegenstehen. Wir stellen daher im folgenden die Minimalanforderungen mit Verweisen auf günstigere Lösungen (unter dem Gesichtspunkt der variablen Kosten für Lernzeit, Verarbeitungszeit, Speicherbedarf u.ä.) dar:

1.1. Studentenplatz

Zum Studentenplatz sind die Geräte für Informationsausgabe (Bildschirm, Fernschreiber) und Adressateneingabe (Tastatur, Lichtgriffel) sowie die Adressatenplatzgestaltung zu zählen.

1.1.1. Dialog-Ein-/Ausgabeeinheit

1.1.1.1. Ausgabe

Beim CET fallen folgende Ausgaben an:

- längere alphanumerische Zeichenketten, z.B.: Erklärung der Parametereingabe;
- Graphiken, z.B.: Darstellung von fixen und variablen Kosten sowie der Erlöskurve in einem Koordinatenkreuz bei der Break-even-Analyse;
- Tabellen, die ohne bedeutenden Verlust an Anschaulichkeit nicht in mehrere Teile zerlegt werden können, z.B.: Simplextableau;

Bei der Ausgabe müssen folgende Bedingungen erfüllt werden:

- Zeichenvorrat

Es muß ein Zeichenvorrat vorhanden sein, der zur Darstellung verbaler Texte geeignet ist. (Die Testversion des CET läuft z.B. auf einem Bildschirm, der keine Ausrufe-, Frage-

und Anführungszeichen besitzt.) Zum Schreiben mathematischer Formeln wären zwar Hochzahlen, Indizes, griechische Buchstaben, Bruchstriche und andere Sonderzeichen wünschenswert, eine Darstellung in FORTRAN-Notation ist aber eindeutig und mit einiger Übung auch ausreichend lesbar.

- Ausgabegeschwindigkeit

Die Ausgabe soll wenigstens so schnell erfolgen, daß der Mensch in seiner Lesegeschwindigkeit nicht beschränkt wird (mindestens 40 Zeichen pro Sekunde bei mäßigem Lesetempo), da Totzeiten die Lernzeit erhöhen und den Lernenden frustrieren.

- Ausgabemächtigkeit

Betriebswirtschaftliche Problemstellungen benötigen ein Ausgabegerät, das es erlaubt, dem Studenten ca. 5000 Zeichen gleichzeitig darzubieten. Auf diese Größenordnung kommt man, wenn man den üblichen Umfang betriebswirtschaftlicher Berichte berücksichtigt. Sie wird auch dadurch bestätigt, daß an der Sloan School des MIT Bildschirme mit 50 Zeilen à 80 Zeichen für ähnliche Problemstellungen wie im CET eingesetzt werden¹⁾. Auch Hedberg gibt an²⁾, daß die üblichen alphanumerischen Bildschirme mit einer Kapazität von 1000 Zeichen für betriebswirtschaftliche Dialoge unzureichend sind.

- Darstellung von Graphiken

Da bei den Versuchen mit dem CET nur alphanumerische Bildschirme zur Verfügung standen, mußten Graphiken aus Sonderzeichen und Buchstaben zusammengestellt werden (s. die Graphiken in den Beispielen). Dies hatte bei ungünstigen Datenkonstellationen oft nahezu unverständliche Abbildungen zur Folge, die man in Analogie zu verdampftem Text³⁾ als

1) Vgl. Rockart, J.F., o.T., a.a.O., S. 15.

2) Hedberg, B., a.a.O., S. 81 f.

3) Weltner, K., Informationstheorie und Erziehungswissenschaft, Quickborn 1970, S. 63 f.

verdampfte Bilder bezeichnen könnte. Eine Erhöhung der Ausgabemächtigkeit von 1000 (bei den Versuchen) auf 5000 Zeichen brächte zwar eine wesentliche Verbesserung der Anschaulichkeit, doch kann nur eine vollgraphische Dialogstation eine befriedigende Lösung bieten, da die Darstellung stetiger Funktionen durch alphanumerische Zeichen stets eine Minderung der Genauigkeit und der Übersichtlichkeit zur Folge hat.

Bei Verwendung eines vollgraphischen Bildschirms wäre es auch möglich, Formeln in üblicher mathematischer Schreibweise darzustellen und auf die FORTRAN-Notation zu verzichten.

- Steuerung audiovisueller Medien

Bei einer größeren Anzahl von RUU-Anwendungen der letzten Zeit wurden Kombinationen mit Ausgabegeräten wie Tonband, Film, Diaprojektor u.ä.m. versucht. Ein Einsatz dieser Medien für das CET kann aber kaum zu einer Kostensenkung führen, da das CET primär Ausgaben mit variablem Inhalt hat und somit diese Medien nur sehr selten einsetzen kann.

Aus den bisher genannten Anforderungen läßt sich die Frage, ob ein Bildschirm oder ein Fernschreiber als Ausgabeeinheit verwendet werden soll, nicht eindeutig klären. In Tab. 2 sind die wichtigsten Nachteile der beiden Medien aufgezählt.

Vollgraphischer Bildschirm	Fernschreiber
Hohe Gerätemiete (bzw. hoher Kaufpreis) Verlust des alten Bildschirminhaltes bei Bildwechsel	Niedrige Ausgabegeschwindigkeit Lärmbelästigung Schwierige Eingabekorrektur Fehlen der (echten) graphischen Darstellung

Tab. 2

Jedes der beiden Medien weist bei der derzeitigen technischen Realisierung noch schwerwiegende Mängel auf. Eine Entscheidung für das eine oder andere hängt sehr stark von der dominierenden Ausgabeart ab. Die Lösung liegt in einem vollgraphischen Bildschirm mit Hardcopy-Vorrichtung¹⁾. Dies ist derzeit zwar eine sehr teure Alternative, für die Zukunft kann man sich allerdings von den Plasma-Displays eine starke Kostensenkung erwarten. Bitzer und Johnson geben einen Kaufpreis von 1500 bis 3000 Dollars für ein Terminal an, das neben dem Plasma-Display auch noch einen Diasortierer und einen Tonsortierer umfaßt²⁾. Für die Erstellung von Hardcopies kann man ebenfalls mit besseren und billigeren Geräten als den derzeit auf dem Markt befindlichen rechnen.

Für betriebswirtschaftliche Probleme ist besonders unter Berücksichtigung der zu erwartenden Hardwareentwicklung einem vollgraphischen Bildschirm der Vorzug zu geben.

1.1.1.2. Eingabe

Die Eingabe beim CET umfaßt

- die Auswahl von Alternativen,
- Bestimmung von Parametergrößen und
- alphanumerische Zeichenketten im Rahmen eines verbalen Dialogs (s. z.B. Lexikon-Strategie) oder als Befehle.

Abgesehen von Tasten zur einfachen Korrektur der Eingabe und einer Übertragungstaste genügt eine gewöhnliche Schreibmaschinentastatur. Ein Lichtgriffel bringt keine wesentliche Erleichterung.

1) Vgl. dazu auch Morton, M.S.S., Management-Entscheidungen im Bildschirmdialog, Essen 1972, S. 145.

2) Bitzer, D.L., Johnson, R.L., a.a.O., S. 11.

1.1.2. Umgebung des Terminals

Die Versuche mit dem CET wurden im Terminal-Raum des Rechenzentrums der Universität Erlangen-Nürnberg durchgeführt. Es standen fünf Bildschirme neben mehreren Fernschreibern und zwei Lochern in demselben Raum. Die Umgebung der Ausgabestationen war somit verhältnismäßig laut. Nach Ansicht der Studenten¹⁾ beeinträchtigte dies den Lernfortschritt stark. Für die Effizienz des CET ist es daher notwendig, den Studentenplatz möglichst gegen Störungen der Umwelt abzuschirmen.

1.2. Rechner

1.2.1. Internspeicherkapazität

Die Anforderungen an die Größe des Internspeichers werden

- durch die im CET behandelten Probleme,
- die Anzahl der gleichzeitig daran angeschlossenen Adressatenplätze und
- die Konzeption des CET-Systems

bestimmt.

Für den Speicherbedarf der Rechenroutinen können keine genauen Angaben gemacht werden. Im derzeit realisierten CET-Kurs über Prognoserechnung sind Programme mit einem Speicherbedarf zwischen ca. 1 und 40 KB enthalten.

Die Anzahl der Benutzer hat vor allem dann einen Einfluß, wenn mehrere speicherintensive Rechenroutinen gleichzeitig abgearbeitet werden sollen.

Der Speicherbedarf wird aber auch durch den Systemteil bestimmt, der resident definiert ist.

1) Vgl. dazu Teil 2, Punkt 5.

Die Größe des benötigten Internspeichers ist somit sehr flexibel und kann nur für einen speziellen Anwendungsfall bestimmt werden. Man kann aber allgemein sagen, daß bereits ein Internspeicher mit ca. 16 bis 32 KB zur Realisierung eines großen Teiles möglicher CET-Anwendungen ausreicht.

1.2.2. Rechengeschwindigkeit

Dialogprogramme stellen keine besonderen Forderungen an die Rechengeschwindigkeit. Schwierigkeiten können sich nur ergeben, wenn mehrere Dialogbenutzer gleichzeitig mit komplexen Rechenroutinen benötigt werden. Für alle anderen Aufgaben könnte ein Kleinrechner verwendet werden. Abb. 7 zeigt, wie ein Kleinrechner mit einem Großcomputer verbunden werden kann.

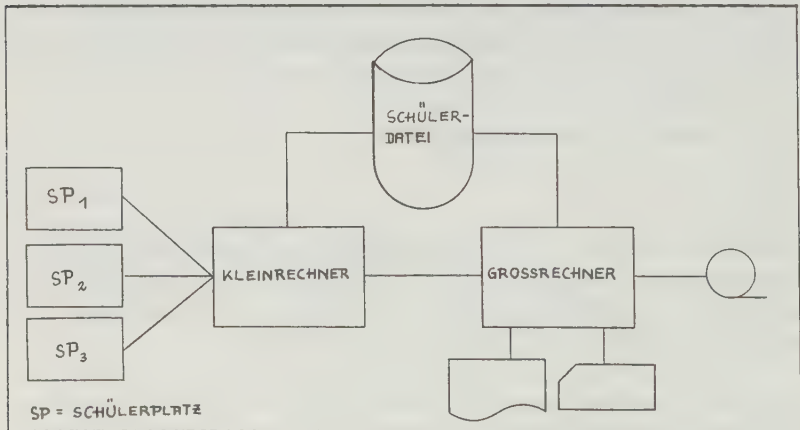


Abb. 7: Koppelung eines Kleinrechners mit einem Großrechner

Die Schülereingabe wird vom Kleinrechner analysiert und in der Schülerdatei gespeichert. Muß eine Rechenroutine bearbeitet werden, die die Kapazität des kleinen Rechners überschreitet, gibt er diese zur Durchrechnung an den Großcom-

puter. Dieser kann auch die Datensicherung von Platte auf Band sowie die Erstellung etwaiger Outputs über den Schnelldrucker übernehmen. Die Ausgabe an den Studenten kann wieder der Kleinrechner ausführen.

Im Bericht der CERI "The Use of Computers in Higher Education, Perspectives and Policies" wird festgestellt:

"At the same time, small computers are being associated with larger systems in a way which removes most of the arguments about system size. Each of the small computers can draw upon the larger processor to work with extended data bases, to call upon specialized software, and to perform large scale calculations which would consume too much processing time on the small local computer used alone. Each small computer is available to work separately"¹⁾.

Der zentrale Rechner wird durch diese Organisationsform um einen billigen Prozessor erweitert. Diese Zwischenschaltung eines Kleinrechners ist keine unbedingte Voraussetzung für die Verwendung von CET, dürfte aber beim Einsatz für größere Adressatengruppen zu einer starken Kostensenkung führen.

1.3. Externspeicher

Im CET wartet der Rechner den Großteil der Zeit auf die Eingabe des Studenten. Es ist daher sinnvoll, während der Wartezeit des Rechners die Rechenroutinen aus dem Internspeicher auf Großbraumspeicher auszulagern, was bei Time-Sharing-Rechnern mit Hilfe des paging und swapping automatisch geschieht. Dies bedeutet eine große Anzahl von Zugriffen auf Externspeichereinheiten. Eine hohe Übertragungsrate und schnelle Zugriffszeiten sind für einen effizienten Betrieb daher Voraussetzung. Für die Programmspeicherung ist eine Magnettrommel vorteilhaft; es genügt allerdings auch eine schnelle Platteneinheit.

1) OECD CERI (Hrsg.), The Use of Computers in Higher Education, Perspectives and Policies, Technical Report, Centre for Educational Research and Innovation, OECD CERI/CT/70.64, S. 27.

Zur Speicherung der Schülerdaten (Antworten, schülerindividuelle Dateien etc.) ist ebenfalls eine Random-access-Einheit notwendig.

Zur Datensicherung sowie zur Aufbewahrung von Schülerdaten für spätere Auswertungen sind Magnetbänder sinnvoll, da diese entsprechend billigen Speicherplatz zur Verfügung stellen können.

1.4. Ein-/Ausgabe-Peripherie (außer Dialogstationen)

Für die Geräte wie Kartenleser und Schnelldrucker ergeben sich durch das CET keine besonderen Anforderungen.

1.5. Zusammenfassung

Die Hardwareanforderungen bei der Realisierung von CET sind durch das Problem des Ein-/Ausgabemediums gekennzeichnet, das technisch noch sehr schlecht gelöst ist. Es muß versucht werden, Softwarelösungen zu finden, welche die Auswirkungen dieses Hardwaremangels einschränken.

2. Anforderungen an die Software

Unter Software seien hier all jene Programme verstanden, die unter Verwendung eines vorhandenen Betriebssystems die Erstellung vom MMK-Programmen, insbesondere des CET, unterstützen.

2.1. Grundsätzliche Entscheidungen

Vor der Bestimmung der Detailanforderungen müssen grundsätzliche Entscheidungen getroffen werden, die das gesamte System betreffen.

2.1.1. Mehrere Teilsysteme oder allgemeines MMK-System?

Zinn¹⁾ vertritt die Forderung, daß nicht ein einzelnes Unterrichtssystem zur Realisierung beliebiger Strategien geschaffen, sondern Spezialsysteme für eine große Zahl von Lehrmethoden dem Programmierer zur Verfügung gestellt werden sollen.

Derartige isolierte Teilsysteme würden zwar die Verwirklichung eines großen Teiles des CET erlauben, verhinderten aber eine Integration der einzelnen Lehrmethoden, die einen wesentlichen Beitrag zur Wirksamkeit des Unterrichts bringt²⁾. Das CET sollte nicht in verschiedenen Systemen, sondern in einem einzigen realisiert werden, das es aber erlaubt, für die verschiedenen Strategien Makros zu erstellen, um so strategienbedingte, ständig wiederkehrende Befehlsgruppen einmal zu programmieren und bei Bedarf wieder aufrufen zu können.

Wird ein System geschaffen, das einerseits die zur Abwicklung eines allgemeinen Dialoges notwendigen Befehle (wie eine Simulationssprache für die Simulation) anbietet, andererseits erlaubt, Makros zu definieren, kann auf eine besondere Philosophie (z.B. PU-Philosophie) bei der Erstellung dieses Systems verzichtet und diese dem Autor überlassen werden. Die Vorteile dieser Konzeption liegen

- in der hohen Kompatibilität, da bei Vorhandensein des Grundsystems sämtliche Teilsysteme erlaubt sind,
- in der Integrationsfähigkeit, da die Studentendaten auf

1) Vgl. Zinn, K., a.a.O., S. 55 f.

2) Vgl. dazu Rockart, J.F., Integrated Use, a.a.O., S. 3 f.

- einem gemeinsamen File gehalten werden, und
- in der Flexibilität, da es für den Autor möglich ist, durch Makros spezielle "Sprachen" für neue Strategien zu schaffen.

2.1.2. Compilierendes oder interpretierendes System?

Die Frage, ob die CET-Programme interpretierend oder kompiliert verarbeitet werden sollen, ist primär ein Problem der Rechenzeit, hat darüber hinaus aber auch einen Einfluß auf den Programmierkomfort.

Ein Rechenzeitvergleich für ein Sortierprogramm in den Sprachen PLANIT, EDI (ein am Rechenzentrum der Universität Erlangen-Nürnberg erstelltes Tischrechnersubsystem innerhalb des Teilnehmersystems RESPOND¹⁾) und FORTRAN zeigte, daß die interpretierenden Systeme PLANIT und EDI um zwei bis drei Zehnerpotenzen langsamer als das kompilierend arbeitende FORTRAN sind²⁾.

Auch im Rahmen des CET wurden Vergleiche zwischen den Bearbeitungszeiten in FORTRAN und PLANIT angestellt, die das oben erwähnte Ergebnis bestätigen³⁾.

Das Rechenzeitverhalten von semiinterpretativen Systemen, die mit einem Precompiler einen internen Programmcode erzeugen, ist zwar wesentlich günstiger als bei einer vollinterpretativen Sprache, jedoch schlechter als beispielsweise bei FORTRAN. Zudem fällt durch die Precompilation ein Teil der Vorteile eines interpretierenden Systems weg.

Wenn auch die ausgewählten Programmbeispiele u.U. nicht vollständig repräsentativ sind, so ergeben sich doch deut-

1) Wolf, F., Seybold, G., EDI: Beschreibung eines einfachen Dialog-Interpreters für die CD 3300, Universität Erlangen-Nürnberg, Rechenzentrum, Interner Arbeitsbericht Nr. 26 Erlangen 1973.

2) Dieser Vergleich wurde von C. Endres durchgeführt.

3) Vgl. dazu Teil 3, Punkt 2.1.4.

liche Hinweise auf das Rechenzeitverhalten der beiden verschiedenen Verarbeitungsarten. Bei einem compilierenden System ist zwar die Übersetzungszeit jedem Lauf eines Programmes anteilig zuzurechnen, doch ist gerade bei rechenintensiven Programmen wie im CET die Verarbeitung eines compilierenden Programmes wahrscheinlich wirtschaftlicher und daher dem compilierenden System der Vorzug zu geben.

2.1.3. Maschinenabhängiges oder -unabhängiges System?

CET-Programme sind in Erstellung und Austestung sehr teuer und müssen daher einem großen Adressatenkreis zugänglich gemacht werden. Dies ist bei der Verschiedenheit der derzeit benutzten Datenverarbeitungssysteme nur dadurch möglich, daß die Programme in einem stark verbreiteten Programmiersystem wie FORTRAN geschrieben werden. Muß dieses aber erweitert werden, so geht die Kompatibilität verloren, wenn die neuen Sprachelemente nicht über einen Metacompiler (z.B. in ein FORTRAN-Programm) umgewandelt werden können. Es ist derzeit nicht möglich, eine neue Sprache oder eine Erweiterung einer bestehenden zu schaffen, die allgemein eingesetzt werden kann, da erst längere Versuche mit neuen Sprachelementen deren Nützlichkeit zeigen können. Es muß versucht werden, über individuelle MMK-Versuchssysteme zu Standardsprachelementen zu gelangen, welche die Grundlagen einer allgemeinen MMK-Sprache bilden können.

Es sollte jedoch schon derzeit ein maschinenunabhängiges Dialogsystem erstellt werden, das es erlaubt, Teachware zwischen einer großen Zahl von Benutzern auszutauschen¹⁾.

2.1.4. Isolierter Adressatenplatz oder Möglichkeit zum Zusammenschluß von Adressatenplätzen?

Ein Teil der CET-Strategien verlangt einen vollständig individuellen Dialog mit dem Studenten, da dieser einen

1) Vgl. dazu auch Bode, A., Dütting, M., Computer-Unterstützter Unterricht - Probleme, Autorensprachen, Systemvergleiche, Dokumentation, Diplomarbeit Karlsruhe 1972, S. 198.

wesentlichen Einfluß auf den Lernerfolg hat.

Auf der Lernzielstufe 2, also bei der Anwendung von Modellen, soll der Student praxisähnliche Bedingungen vorfinden. Dazu gehören auch das Arbeiten im Team und die Einführung des Konkurrenzgedankens. Der Teamgedanke kann durch die Bildung von Studentengruppen verwirklicht werden, die mit einem gemeinsamen Terminal arbeiten. Sollen aber mehrere Personen in verschiedenen Funktionen an demselben Problem arbeiten, so müssen mehrere Studentenplätze einem Dialogprogramm zugeordnet werden können.

2.2. Systemaufbau

2.2.1. Baukastenprinzip

Um ein System für möglichst viele Datenverarbeitungsanlagen verwenden zu können, muß es nicht nur von der Sprache her verträglich, sondern auch der jeweiligen Rechnerkapazität anpaßbar sein.

Dieses Problem soll durch ein Baukastensystem gelöst werden, das es erlaubt, bei der Systemgenerierung auf einen Rechner nur jene Systemteile aufzunehmen, die bei der speziellen Anwendung gebraucht werden und die Rechnerkapazität nicht überschreiten.

2.2.2. Integration der Strategien

Die verschiedenen Strategien des Systems sollen dem Adressaten stets zur Verfügung stehen, um so dem Lerner den jeweils besten Lernweg zu erlauben. Er soll dabei stets auf seine persönlichen Dateien zugreifen können.

Die Lernersteuerung, die in den derzeit realisierten Unterrichtssystemen nur in Ansätzen zu finden ist, kann erst

durch eine Verarbeitung mehrerer Strategien in demselben System voll wirksam werden.

Die Lerndatenaufzeichnung muß sowohl den Weg innerhalb einer Strategie als auch die Sprünge zwischen den Strategien erfassen, da diese Daten für den Autor gerade in einem Lehrmethoden-"Mix" sehr wichtig sind.

2.2.3. Routinedefinitionen

Im hier beschriebenen CET-System sollen unterschiedliche Dialogformen möglichst einfach realisierbar sein. Dies bedeutet, daß innerhalb des Systems durch Routinen "Quasi-Sprachen" gebildet werden können, die eine besonders einfache Programmierung durch den Autor erlauben.

So soll man beispielsweise eine "Sprache" zur Erstellung eines Lexikons erstellen können, in welcher der Programmator die Lexikoninhalte nur noch in einem vorgegebenen Format eingeben muß und die "Sprache" die Ablauflogik erstellt. Derart einfache Sprachen werden von Zinn als "data formats" bezeichnet¹⁾.

Aber auch für komplexere Dialogstrukturen können Routinen gebildet werden, welche die Programmierung erleichtern. So könnte beispielsweise eine Routine geschaffen werden, die Möglichkeiten wie eine PU-Sprache bietet.

2.2.4. Trennung von Text, Struktur und Rechenroutinen

Um eine möglichst einfache und übersichtliche Programmierung und Dokumentation zu erreichen, empfiehlt es sich, Texte, Ablaufstrukturen und Rechenroutinen gesondert ein-

1) Vgl. Zinn, K., a.a.O., S. 50 f.

zugeben. In PUMA und im TS wird bereits eine Trennung von Text und Struktur vorgenommen. Beim CET, das durch eine Verfahrensbank gekennzeichnet ist, bringt die getrennte Eingabe der Rechenroutinen einen weiteren Gewinn an Übersichtlichkeit¹⁾.

Die Dialogstruktur legt den Ablauf der Informationsdarbietung und den Aufruf von Rechenroutinen fest. In ihr werden auch die Studenteneingabe, Datenübergabe an die Rechenroutinen sowie die Ausgabe von Ergebnissen und Texten definiert.

Die Rechenroutinen umfassen nur die Programme zur Durchrechnung von Algorithmen. Sie nehmen die Studenteneingabe und Datenbereiche von den Files über die Struktur entgegen und geben die Ergebnisse an die Struktur zurück.

Diese Trennung bietet neben den bereits erwähnten Vorteilen die Möglichkeit, vorhandene Standardprogramme, die normalerweise keine Ein-/Ausgabeoperationen enthalten, in eine Verfahrensbibliothek aufzunehmen.

2.3. Systemteile

Als Systemteile seien hier Programmelemente bezeichnet, die bei der Erstellung von Unterrichtsdialogen häufig benötigt werden und daher in einem Dialogsystem standardisiert gelöst und für sämtliche Strategien verfügbar sein müssen.

Die Anforderungen werden im folgenden abstrakt formuliert und dann mit den Realisierungsmöglichkeiten in PLANIT und PUMA verglichen. Zudem werden sie auch den Lösungen gegenübergestellt, die im Versuchssystem des CET (TS) gewählt wurden. Weiterhin wird angegeben, welche der noch nicht realisierten Forderungen in der geplanten Verbesserung des TS berücksichtigt werden sollen.

1) Vgl. Teil 3, Punkt 2.4.2.

Eigenschaften, die bereits in den derzeit gebräuchlichen Unterrichtssystemen zur Selbstverständlichkeit geworden sind, werden wegen der Kürze der Darstellung häufig nicht aufgeführt (z.B. die Möglichkeit zur Abfrage der Tageszeit). Die Anforderungen beziehen sich meist auf die Verwendung von Bildschirmen, da diese bezüglich des Ausgabeformates engere Restriktionen als Fernschreiber besitzen.

2.3.1. Struktur

2.3.1.1. Aufgaben

Die Struktur steuert den Ablauf von Operationen wie die Ausgabe, Eingabe, Eingabeanalyse, den Zugriff auf Platte oder Band, die Ausgabe über Drucker, Plotter und andere Geräte und den Aufruf von Rechenroutinen.

In ihr können auch Rechenoperationen zur Ablaufsteuerung durchgeführt werden.

Der Dialogablauf von Unterrichtsprogrammen ist selten rein linear. Er besitzt meist eine große Zahl von Verzweigungen, die von der Adressatenantwort, den Lerndaten und anderen Faktoren abhängen. Der Autor muß Befehle zur Verfügung haben, die eine möglichst einfache Definition solcher Strukturen gestatten. Die Programmiersprache für die Struktur sollte ihm folgende Möglichkeiten bieten:

- Unterprogrammtechnik mit und ohne Parameterübergabe,
- Definition von Makrobefehlen,
- Drei-Weg-Verzweigung mit variablen Verzweigungspunkten,
- logische Verzweigung mit variabler Sprungadresse,
- Sprungbefehl mit variabler Sprungadresse,
- berechneter Sprungbefehl,
- Laufschleife,
- Restart-Points,
 - automatisch
 - autordefiniert

- Definition von Statement-labels,
- automatische Zeilennumerierung und
- folgende Datentypen: String, Character, Real, Integer, Gleitkomma.

2.3.1.2. Realisierung in vorhandenen Systemen

Bei den RPU-Sprachen können zwei prinzipielle Sprachstrukturen unterschieden werden:

- die Frame-Struktur und
- die Befehlsstruktur.

Sprachen, die hauptsächlich zur Präsentation von Programmierem Unterricht geschaffen wurden, verwenden meist den Frame-Aufbau. In einem frame sind Informationsausgabe, Frage, Eingabe, Antwortanalyse und darauf basierende Aktionen zu einer Lerneinheit zusammengefaßt. Anhand von PLANIT wurde die Eignung dieses Aufbaues für das CET untersucht.

2.3.1.2.1. PLANIT

In PLANIT werden vier Frame-Arten unterschieden,

- das question,
- multiple choice,
- decision und
- programming frame.

Question und multiple choice frame können als Lerneinheiten, wie sie oben beschrieben wurden, bezeichnet werden, während das decision frame speziell zur Steuerung des Ablaufes und das programming frame zur Definition von Rechenroutinen bestimmt sind.

In PLANIT wird somit die Mikrostruktur eines Lernprogrammes innerhalb des question und multiple choice frame's und die Makrostruktur eher in decision frames festgelegt.

Durch den Frame-Aufbau tritt eine gewisse Standardisierung des Dialogablaufes ein. Diese bringt einerseits den Vorteil der leichteren Programmierbarkeit von reinen Lernprogrammen, erschwert aber andererseits die Realisierung von Dialogstrategien, die nicht in diesem Konzept vorgesehen wurden. Das CET benötigt für seine Lehrstrategien einen Aufbau, der sich stark von dem von Unterrichtsprogrammen unterscheidet. Versuche, CET-Programme in einer Frame-Struktur zu erstellen, haben gezeigt, daß Programmiersprachen dieser Art einen größeren Aufwand als beispielsweise FORTRAN verursachen.

Eine Besonderheit von PLANIT ist die standardisierte Unterstützung der extrinsischen Programmierung durch die Bereitstellung folgender "Conditionals" zur Lernwegbestimmung:

MINUTES: Zeitbedarf des Adressaten zur Durcharbeitung bestimmter Programmteile

SEEN: Binärvariable zur Anzeige, ob ein bestimmtes frame angesprungen wurde

USED: Speichert, ob der Student eine Calc-Funktion benutzt hat

RIGHT, WRONG, +, -: Geben Auskunft über die Richtigkeit der Antworten.

Zur Realisierung von CET-Programmen sind diese für den PU wertvollen Programmierhilfen nur beschränkt verwendbar, da das CET weitestgehend auf eine Steuerung des Studenten innerhalb einer Lektion verzichtet.

Eine Laufschleife im Sinne des DO in FORTRAN ist in PLANIT nicht vorhanden. Die Unterprogrammtechnik ist durch die höchstzulässige Übergabe von zehn Werten sehr stark begrenzt. Zudem ist es nicht möglich, Makrobefehle zu definieren. Die Forderung nach einer automatischen Numerierung der Zeilen wird innerhalb der Gruppen nicht erfüllt.

PLANIT kennt nur Real-Daten. Während sich das Fehlen der Integer- und Gleitkommadarstellung bei der CET-Programmierung kaum auswirkt, stört das vollkommene Fehlen von String- und Character-Variablen u.U. stark.

2.3.1.2.2. PUMA

PUMA-Programme werden in frei programmierbare Knoten gegliedert. Durch einen Ausgabebefehl wird ein Knoten beendet und der Folgeknoten festgelegt.

Innerhalb eines Knotens stehen sämtliche von uns geforderten Steuerungsbefehle mit Ausnahme des logischen Vergleiches zur Verfügung, der jedoch leicht durch die Drei-Weg-Verzweigung ersetzt werden kann.

PUMA ist weder rein ~~frame~~- noch rein befehlsorientiert. Dies erleichtert zwar die Programmierung von CET-Kursen gegenüber PLANIT, ist aber nicht so einfach und übersichtlich wie eine rein befehlsorientierte Sprache.

Die Unterprogrammtechnik ist in PUMA ebenso wie in PLANIT unzureichend. Ein Parameteraustausch zwischen Programmen ist nur über einen Plattenfile möglich. Der Autor kann außerdem keine Makrobefehle definieren. Dagegen stört das Fehlen von variablen Sprungadressen bei der Drei-Weg-Verzweigung und beim Sprungbefehl ebenso wie der Mangel einer Laufschleife die Programmierung kaum.

Restart-Points werden automatisch definiert; der Autor kann darauf nur durch Anweisungen an den Schüler Einfluß nehmen.

Von den geforderten Datentypen sind in PUMA Character-, Integer- und Real-Variable erlaubt.

2.3.1.2.3. TS

Das TS hat einen rein befehlsorientierten Aufbau der Dialogprogramme, was nach unseren bisherigen Erfahrungen den Erfordernissen der CET-Programmierung am besten entspricht. Durch die Verwendung des FORTRAN-Compilers sind sämtliche im USASI-FORTRAN vorhandenen Steuerungsmöglichkeiten und Varibalentypen erlaubt. Zusätzlich gibt es im FORTRAN der CD 3300 die Möglichkeit des characterweisen Zugriffs. Forderungen, die darüber hinausgehen, sind bisher nicht erfüllt worden.

2.3.1.2.4. Geplanter Ausbau des TS

In der Erweiterung des TS soll vor allem die Speicherung von Restart-Points vorgesehen werden. Makros sollen in der ersten Ausbaustufe noch durch Unterprogramme ersetzt werden. Die Angabe von variablen Sprungadressen ist in FORTRAN nicht möglich und daher im TS auch nicht zu verwirklichen.

2.3.2. Rechenroutinen

2.3.2.1. Aufgaben

Programme zur Durchrechnung von Algorithmen bilden den Schwerpunkt des CET. Der Student muß eine große Anzahl von Verfahren zur Verfügung haben und diese im Dialog wiederholt durchrechnen können. Daraus ergeben sich folgende Forderungen an die Definition und Verarbeitung von Rechenroutinen:

- Möglichkeit zum Einbau vorhandener Rechenroutinen:
Sehr viele der benötigten Verfahren liegen als Bibliotheksroutinen vor. Können diese Programme nahezu unverändert in CET-Kurse übernommen werden, so fallen Programmier- und Testzeiten für diesen sonst sehr aufwendigen Teil der Erstellung von Kursen fast weg. Der CET-Autor muß nur die für das jeweilige Verfahren benötigten

Daten im Strukturprogramm bereitstellen und erhält die Ergebnisse vom Rechenverfahren zurückübermittelt. Dies hat weiter den Vorteil, daß er in der Programmierung nicht so lange ausgebildet werden muß, da komplexe Modelle meist schwieriger als Dialogabläufe zu verkoden sind.

- Rechenfähigkeiten wie in problemorientierter Sprache:
Sind die für den Kurs benötigten Algorithmen nicht in programmierter Form vorhanden, müssen sie unter Benutzung der Möglichkeiten einer problemorientierten Sprache erstellt werden können. Wenn die Möglichkeit geschaffen ist, in dieser Sprache vorhandene Programme anzuschließen, können neue Algorithmen auch in dieser geschrieben werden.
- Hohe Verarbeitungsgeschwindigkeit:
Da das CET die hohe Rechenkapazität des Computers ausnützen will, muß von der Software her auch danach getrachtet werden, daß diese voll zur Verfügung steht. In den üblichen problemorientierten Sprachen ist dies durch die compilierende Verarbeitung hinreichend erfüllt, doch hat APL bewiesen, daß dies auch mit einem semiinterpretativen Verfahren gut erreicht werden kann.
- Möglichkeit, Rechenroutinen als Unterprogramme zu definieren, die von beliebigen Programmen aus aufgerufen werden können:
Rechenroutinen werden oft in verschiedenen Dialogstrukturen benötigt. So kann man beispielsweise die exponentielle Glättung erster Ordnung als eigenes Verfahren zur Erstellung von Prognosen in einem Kurs über Zeitreihenvorhersage benutzen. Sie kann aber auch als Teil des Verweilzeit-Prognoseverfahrens von Langen¹⁾ eingesetzt werden. Es muß daher möglich sein, auf Rechenroutinen aus beliebigen Dialogstrukturen zuzugreifen.

1) Langen, H., Weinthaler, F., Prognose mit Hilfe von Verweilzeitverteilungen, in: Mertens, P. (Hrsg.), Prognoserechnung, a.a.O., S. 113 ff.

2.3.2.2. Realisierung in vorhandenen Systemen

Rechnerunterstützte Unterrichtssysteme stellen für das CET meist nur sehr unzulängliche Rechenmöglichkeiten zur Verfügung. Sie bieten zur Durchführung von Berechnungen oft kaum mehr Kapazität an, als zur Steuerung des Dialogablaufes notwendig ist. Berechnungen müssen zudem in einigen RUU-Sprachen in assemblerähnlicher Form geschrieben werden (vgl. COURSEWRITER III¹⁾, LIDIA²⁾). Ebenso fehlen oft grundlegende Standardfunktionen wie die Quadratwurzel, Potenzierung etc.

2.3.2.2.1. PLANIT

In PLANIT dient der Calculation-Mode (Calc-Mode) der Durchführung von Rechenarbeiten. Er steht sowohl dem Studenten während des Lerndialoges und dem Lehrer bei der Lehrprogrammierung als Tischrechnerfunktion als auch zur Programmierung von Rechenroutinen zur Verfügung. Programme im Calc-Mode werden in einem programming frame erstellt und sind somit fester Bestandteil eines Lernprogrammes. Es stehen FORTRAN-ähnliche Rechenmöglichkeiten bereit.

Der Calc-Mode stellt folgende Operatoren zur Verfügung:

Arithmetische: +, -, *, **(>), /, PROD, SUM
Matrizenoperationen: INVERT, RANK
Schleifenoperator: FOR

Daneben bietet der Calc-Mode eine große Zahl von Funktionen an. Die für das CET wichtigsten werden hier aufgezählt:

RANDOM: Erzeugung rechteckverteilter Zufallszahlen
NORMAL: Erzeugung normalverteilter Zufallszahlen
ABSOLUTE: Absoluter Betrag
SQRT: Quadratwurzel
SIN: Sinus
TAN: Tangens

1) IBM (Hrsg.), Coursewriter III, Version 3, Application Description, IBM Form GH 20-0987-2.

2) Siemens (Hrsg.), Siemens System 4004 LIDIA, Autorensprache für rechnergestützte Unterweisung, Beschreibung, D14/4005.

PLANIT weist für CET-Programme aber folgende wesentliche Nachteile auf:

- In anderen Sprachen erstellte Routinen können nicht angeschlossen werden.
- Die Unterprogrammtechnik ist für Programme mit größeren Übergabebereichen unzureichend. Nur zehn Parameter können mit Hilfe des LINK-Vektors zwischen rufenden und gerufenen Programmen ausgetauscht werden. Dies bedeutet, daß Rechenroutinen, die mehr als zehn Übergabeparameter brauchen, in jedem Kurs als Teil des Hauptprogrammes eingegeben werden müssen.
- Die Rechenzeiten für komplexe Rechenprogramme sind wegen des interpretativen Verarbeitungsmodus sehr hoch. Dies hat einerseits hohe Verarbeitungskosten und andererseits große Antwortzeiten des Systems zur Folge.
- Das Fehlen einer Laufschleife wie das FORTRAN-DO wirkt sich nur auf den Komfort der Programmierung aus.
- Die maximale Frame-Größe - ein Installationsparameter - ist für Rechenroutinen unter Umständen eine kaum zu umgehende Restriktion.
- Ebenso unterliegt die Zahl der vom Autor definierbaren Variablen und Matrizenelemente einer Beschränkung, die bei der Installation von PLANIT festgelegt wird.

2.3.2.2.2. PUMA

Auch PUMA bietet für RUU-Sprachen überdurchschnittliche Rechenfähigkeiten an. Dabei ist vor allem die Rechengeschwindigkeit hervorzuheben, die wegen der compilierenden Verarbeitung ähnlich wie in FORTRAN ist.

PUMA bietet eine FORTRAN-ähnliche Notation, in der Programme beliebiger Länge geschrieben werden können.

Folgende arithmetische Operatoren sind vorhanden: +, -, *, /. PUMA bietet ebenso wie PLANIT eine Reihe von Standardfunktionen an¹⁾.

Es besteht zudem die Möglichkeit, von PUMA aus die Tischrechnerfunktion EDI zu benutzen.

Bei der Erfüllung der Anforderungen sind folgende Mängel festzustellen:

- Unterprogramme können nur in PUMA geschrieben werden, wenn man von der sehr beschränkten Möglichkeit, Assembler- und FORTRAN-Funktionen anzuhängen, absieht.
- Parameterübergabe zwischen Programmen ist nicht vorgesehen. Dieser Mangel kann jedoch durch Auslagern auf Plattenfile umgangen werden.
- Die Anzahl der Variablen ist auf 115 begrenzt. Dies stellt in PUMA die stärkste Beschränkung für die Erstellung von CET-Programmen dar.
- Wie in PLANIT ist auch in PUMA keine Laufschleife wie das FORTRAN-DO möglich.
- Ebenfalls nachteilig für die Erstellung von Rechenprogrammen ist die Begrenzung auf eine einzige arithmetische Operation pro Statement.

PLANIT und PUMA zählen trotz der aufgezählten Nachteile im Rahmen der RUU-Sprachen bezüglich der Rechenfähigkeit zu den leistungsfähigsten Systemen, setzen der Erstellung von Rechenroutinen für das CET aber dennoch zu enge Grenzen.

2.3.2.2.3. TS

Durch die Verwendung von FORTRAN im TS sind sämtliche Forderungen erfüllt, die an den Programmierungskomfort für

1) Da die Standardfunktionen gerade zum Zeitpunkt des Abschlusses dieser Arbeit in PUMA implementiert wurden, kann hier noch keine Liste davon wiedergegeben werden.

die Rechenroutinen zu stellen sind (s. FORTRAN-Programm in Teil 3, Punkt 2.4.2.).

2.3.2.2.4. Geplanter Ausbau des TS

Ein Ausbau im TS ist hierfür nicht notwendig.

2.3.3. Ausgabeaufbereitung

2.3.3.1. Aufgaben

Die Aufbereitung der Ausgabe ist eine der zeitaufwendigsten Tätigkeiten bei der CET-Programmierung. Während für Textkonversen in den meisten Unterrichtssystemen einfache Makros zur Verfügung stehen, ist es bereits schwieriger, variable Zahlenwerte auszugeben, insbesondere wenn diese mit Texten kombiniert werden sollen. Sehr aufwendig ist die Ausgabe von Tabellen und Graphiken. Wegen der hier gebotenen Kürze sei primär die Bildschirmausgabe behandelt. Die Unterschiede in den Anforderungen bei Verwendung von Bildschirm- oder Fernschreibstationen ergeben sich vor allem aus den unterschiedlichen Hardwarerestriktionen. Bei der anschließenden Darstellung der Anforderungen an die Unterstützung der Ausgabe sei jedoch zugrunde gelegt, daß das geforderte RUU-System die Restriktionen der jeweils verwendeten Dialogstationen berücksichtigt.

2.3.3.1.1. Bestimmung des Ausgabemediums

- durch die Hardware des vorhandenen Systems

Da ein Dialogsystem einem breiten Adressatenkreis zugänglich gemacht werden soll, muß es sich möglichst gut an verschiedene Hardwarekonfigurationen anpassen können. Sind nur Fernschreib- oder nur Bildschirmterminals vorhanden, soll das System die jeweils vorhandenen Ausgabegeräte bedienen können.

- durch die Lernenden

Die Entscheidung, welchem Ausgabemedium der Vorzug zu geben ist, wird oft durch Faktoren bestimmt, die nicht vom Dialogprogramm aus generell erfaßbar sind. Es ist daher in vielen Fällen günstig, dem Lernenden die Medienauswahl zu überlassen.

- durch den Programmautor

Ist es für bestimmte Ausgaben sinnvoll, das Ausgabemedium vom Dialogprogramm aus zu bestimmen, so muß auch dies möglich sein. Dieser Fall kann beispielsweise dann eintreten, wenn der Ausgabebereich der vorhandenen Bildschirme für bestimmte Formate nicht ausreicht. Der Programmautor muß neben den Terminals aber auch periphere Geräte wie den Schnelldrucker benutzen können. Dies ist z.B. dann wichtig, wenn lernerindividuelle Arbeitsunterlagen ausgegeben werden müssen und die Dialog-Peripherie dazu nicht geeignet ist.

- für den Terminalverbund

Sollen mehrere Lernende gleichzeitig an einem CET-Programm arbeiten, also beispielsweise an einem On-line-Spiel teilnehmen, ist es erforderlich, mehrere Datenendgeräte von einem interaktiven Programm aus zu bedienen. Dazu muß der Autor logische Geräteummern angeben, d.h. unabhängig von der physischen Terminalnummer aufgrund des Benutzernamens oder eines anderen Kriteriums die Funktion eines bestimmten Adressaten festlegen können.

2.3.3.1.2. Ausgabearten

2.3.3.1.2.1. Textkonserven

Der Befehl zur Ausgabe von Textkonserven muß die Möglichkeit bieten, einen oder mehrere Texte gleichzeitig auszugeben. Dies hat den Vorteil, daß Teile einer Ausgabe, die

öfter vorkommen, nur einmal eingegeben und gespeichert werden müssen. Den Textkonserven sollen Namen, die wie Variablenbezeichnungen verwendet werden können, zugeordnet sein. Für bestimmte Probleme ist es notwendig, die Ausgabe vollständig variabel aus verschiedenen Zeichenketten zusammenzusetzen. Es ist dann nötig, derartige Texte mit Hilfe von Textsubstitution und Konkatenation bilden zu können.

2.3.3.1.2.2. Variable Inhalte

Das CET verlangt bei der Ausgabe oft eine Verknüpfung von Textkonserven mit im Programm errechneten Werten. Die Einfachheit der Lösung dieses Problems hängt vor allem von der Art der Angabe des Ausgabeformates (Plazierung auf dem Bildschirm, Stellenanzahl) und der Zuordnung von Variablen ab.

2.3.3.1.2.2.1. Skalare

Für eine einfache Definition des Outputs sollten daher die Variablennamen an jenen Stellen des Textes angegeben werden können, auf welche die im Programm errechneten Werte gesetzt werden sollen. Die Stellenanzahl müßte man dann gesondert bestimmen.

Folgender Output sei zur Illustration dieser Ausgabeart dargestellt:

FUER DEN ZEITRAUM VON DER PERIODE 1 BIS 16 ERHALTEN SIE MIT
HILFE DER EXPONENTIELLEN GLAETTUNG ERSTER ORDNUNG BEI EINEM
ALPHA VON 0.3 DEN WERT 1713.2.

Diese Art der Ausgabedefinition ist besonders für die Darstellung von wenigen variablen Werten geeignet.

2.3.3.1.2.2.2. Vektoren

Sollen jedoch mehrere Vektoren gleichzeitig dargestellt werden, ist die Verwendung einer Tabellenroutine notwendig, deren Aufruf im Textteil erfolgen soll. Diese Routine muß folgende Möglichkeiten bieten:

- Mehrere Vektoren sind gleichzeitig darstellbar.
- Sie können eine unterschiedliche Anzahl von Elementen be-sitzen,
- die horizontal oder vertikal angeordnet werden können.
- Die Platzierung der Werte auf dem Ausgabemedium soll auto-matisch geschehen.
- Die Tabelle muß in eine Textkonserve eingelagert werden können, so daß man Zeilen- und Spaltenüberschriften sowie einen Begleittext ausgeben kann. Doch auch bei der Defi-nition der Tabelle selbst soll der Autor Text eingeben können.
- Bei der Verwendung eines Bildschirms als Ausgabemedium muß die Tabellenroutine selbständig eine Bildfolge schaf-fen, wenn die Ausgabe nicht in einem einzigen Bild Platz hat.
- Die Formatierung der Ausgabeelemente soll generell oder individuell möglich sein.

Gewöhnlich muß nur eine Teilmenge der oben geforderten Mög-lichkeiten vorhanden sein, um den Anforderungen eines kon-kreten Falles gerecht zu werden. Muß der Autor aber eine einfache Tabelle in einer Routine definieren, die für kom-plexe Fälle vorgesehen ist, so erfordert dies wesentlich mehr Eingabearbeit, als bei einer Routine für einfache Standardfälle. Sieht man dagegen nur eine sehr einfache Tabellen-Routine vor, so wird die Erstellung komplexer Ta-bellen umständlich. Es ist daher sinnvoll, wenn dem Autor mehr als eine Routine zur Verfügung steht.

Wir beschreiben im folgenden an einem Beispiel je eine Mög-lichkeit einer komplexen und einer einfachen Routine, an denen die oben aufgeführten Forderungen näher erläutert werden¹⁾.

1) Die hier verwendete Notation ist nicht als die einzige oder beste Lösung zu verstehen, soll aber einen Eindruck vermitteln, wie einfach es sein soll, Tabellen auszuge-ben.

Beispiel: Komplexe Tabelle

Eingabe im Textteil:

TEXTAB1 (Namen der Texteinheit)

TABA (Tabellenroutine)

O BEN /MONAT/UM-/VORHER-/

O BEN / /SATZ/SAGE/

WERTE X(I) SX(J) FUER (I=1,4)(J=2,5)

RUND X,2; SX,2

LINKS /JANUAR/FEBRUAR/MAERZ/APRIL/MAI

RECHTS /□□□□□□□□ /

TEXTX (Namen der Texteinheit)

AUS FOLGENDER TABELLE KOENNEN SIE IHRE UMSATZ- UND PRO-
GNOSEDATEN ENTNEHMEN:

TEXTY

BITTE GEHEN SIE MIT SEND WEITER!

Der Ausgabebefehl

BILD TEXTX TEXTAB1 TEXTY

bringt folgenden Output:

AUS FOLGENDER TABELLE KOENNEN SIE IHRE UMSATZ-
UND PROGNOSEDATEN ENTNEHMEN:

MONAT	UM- SATZ	VORHER- SAGE
JANUAR	133.22	
FEBRUAR	135.10	133.22
MAERZ	138.79	137.52
APRIL	142.34	141.84
MAI		145.93

BITTE GEHEN SIE MIT SEND WEITER!

Erläuterungen zum Beispiel:

Mit OBEN werden die Spaltenüberschriften definiert, die mehrere Zeilen umfassen können. Gibt man die Anweisung zur Bestimmung der Vektoren (WERTE) nach OBEN oder UNTEN,

so erscheinen diese als Spaltenvektoren. Nach LINKS oder RECHTS bewirkt sie eine horizontale Anordnung. Durch die Angabe von Leerstellen in RECHTS wird der Platz für eine vierte Spalte vorgesehen, in der aber nichts steht. Damit werden die anderen Spalten nicht so weit auseinandergerissen. Die Angaben nach FUER sind nicht als Laufschleifen im Sinne von FORTRAN zu verstehen. Sie geben damit die Zeilen an, in denen die Variablen ausgedruckt werden. Es muß auch eine Angabe von Schrittweiten und eines Beginnindex möglich sein, wenn dieser nicht gleich eins ist.

Eine Tabellenroutine für einfachere Probleme könnte wie folgt aussehen:

Beispiel: Einfache Tabelle

Eingabe:

TEXTTAB15 (Namen der Texteinheit)

TABB (Tabellenroutine)

X(I), SX(J) FUER (I=1,4)(J=2,5)

Der Ausgabebefehl

BILD TEXTX TEXTTAB15 TEXTY

bringt folgenden Output:

AUS FOLGENDER TABELLE KOENNEN SIE IHRE UMSATZ-
UND PROGNOSEDATEN ENTNEHMEN:

133.22	
135.10	133.22
138.79	137.52
142.34	141.84
	145.93

BITTE GEHEN SIE MIT SEND WEITER!

2.3.3.1.2.2.3. Darstellung von Graphiken

Die Graphik wird wie die Tabelle sehr oft benötigt. Im CET-System soll daher eine Graphikroutine vorhanden sein, die es erlaubt, nahezu alle Wünsche zu berücksichtigen, ohne daß dadurch die Eingabe für diese Routine allzu komplex wird. Um dieser Anforderung vollständig gerecht zu werden, ist es wie bei der tabellarischen Ausgabe notwendig, mehrere Graphikprogramme zu schaffen.

Folgende Forderungen sind an eine Routine zu stellen, die für nahezu alle Fälle ausreicht:

- Alle vier Quadranten des Koordinatenkreuzes müssen darstellbar sein.
- Der Maßstab für die Koordinaten muß automatisch festgelegt werden. Bei Bedarf muß ihn aber auch der Student oder der Programmator bestimmen können.
- Der Autor muß die Möglichkeit besitzen, einzelne Funktionen von der Maßstabberechnung auszunehmen.
- Das Achsenkreuz muß so auf dem Bildschirm angeordnet werden, daß zur Darstellung der Funktionen der ganze Bildschirm zur Verfügung steht.
- Der Schnittpunkt der Koordinatenachsen muß so gewählt werden, daß die Funktionen den ganzen Bildschirm ausnutzen.
- Mehrere Funktionen müssen gleichzeitig gezeigt und unterschiedlich große Definitions- und Wertebereiche haben können.
- Die Achsen müssen automatisch mit dem errechneten Maßstab beschriftet werden.
- Durch Angabe eines Parameters muß es möglich sein, auch ein Histogramm auszugeben.
- Wie die Tabelle soll man auch die Graphik in eine Textkonserve legen können.
- Die Anzahl der Zeilen und Spalten, die für die Graphik verwendet werden soll, muß auf Wunsch durch den Autor bestimmbar sein. Dies ist vor allem bezüglich der Zeilenanzahl bei der Fernschreibausgabe wichtig.

Ist für das System ein vollgraphischer Bildschirm verfügbar, müssen die Graphikroutinen auch für diesen geschaffen werden. Weiterhin sollen Routinen zur Ausgabe von standardmäßig nicht vorgesehene Zeichen (z.B. $\angle$, Bruchstrich, Wurzel) verfügbar sein¹⁾. Für den vollgraphischen Bildschirm müssen ebenso Routinen geschaffen werden, die eine leichte Darstellung von Graphiken wie Entscheidungsbäume oder Netzpläne erlauben. Anforderungen an diese Makros können hier nicht aufgestellt werden, da wir die Experimente nur auf kleinen alphanumerischen Displays durchführen konnten, die für diese Art der graphischen Ausgabe vollkommen ungeeignet sind.

2.3.3.2. Realisierung in vorhandenen Systemen

Die Ausgabeaufbereitung ist in RUU-Systemen hauptsächlich auf die Darbietung konstanter Texte ausgerichtet. Die Darstellung variabler Inhalte wird nur wenig unterstützt.

2.3.3.2.1. PLANIT

In PLANIT gibt es mehrere Möglichkeiten, Text auszugeben. Innerhalb des question- und des multiple choice-frames kann an der dafür vorgesehenen Stelle (Gruppe 2) Text eingegeben werden, der bei Durcharbeitung dieses frame's auf dem Terminal erscheint.

Daneben gibt es eine Reihe von weiteren Möglichkeiten, dem Studenten Texte darzubieten:

f: (feed back): Als unmittelbare Reaktion auf die Antwort des Studenten kann mit f: Text ausgegeben werden.

r: (restart): Hat der Student bei der Eingabe einen Fehler gemacht und soll er seine Antwort wiederholen, kann er mit r: in den Antwortmodus zurückversetzt werden. Dabei kann der Autor eine Anweisung ausgeben.

Gibt er hinter den Kommandos f: und r: keinen Text an, erfolgt eine automatische Meldung des Systems an den Studenten.

1) Vgl. dazu Schmitt, A., Zum Entwurf eines flexiblen und portablen Software-Systems für Rechnergestützten Unterricht, Institut für Informatik der Universität Karlsruhe, Interner Bericht Nr. 7, Karlsruhe 1972.

Gibt er c: allein ein, erhält der Student automatisch die richtige Antwort.

In allen frames kann man mit den Befehlen print und align Texte und variable Werte ausgeben. Print wird hauptsächlich für Text benutzt, in dem auch Variable enthalten sind. Align bietet zu den Möglichkeiten von print noch die Angabe der Stelle, ab der die Variable oder ein Text geschrieben werden soll.

In PLANIT kann nur ein Fernschreiber oder ein sog. tabellarischer Bildschirm verwendet werden. Ein Zusammenschluß mehrerer Terminals in einem Lernprogramm ist nicht möglich.

Die genannten Ausgabemöglichkeiten erlauben es nicht, Strings zu definieren, die beim Output beliebig zusammengestellt werden können. Die relativ große Zahl von Ausgabearten und die Eingabe der Texte gemeinsam mit der Struktur machen das Lernprogramm unübersichtlich. Dies wirkt sich besonders dann aus, wenn nicht Programmierter Unterricht in reiner Form betrieben wird, weil damit die Frame-Einteilung zum Teil dem Programmaufbau nicht mehr entspricht.

Die Ausgabe von Skalaren wird im geforderten Umfang durch print und align unterstützt. Ebenso genügt das align-State-ment in Verbindung mit dem Operator FOR für einfache Tabellen. Komplexe numerische Ausgaben und Graphiken erfordern in PLANIT vom Autor einen hohen Programmieraufwand.

2.3.3.2.2. PUMA

PUMA bietet folgende Möglichkeiten zur Textausgabe:

- Mit dem Befehl DP (display) können Textkonserven ausgegeben werden, die getrennt vom Strukturprogramm auf einem Textfile vorliegen. Der Autor kann zwei Textteile in einem DP-Befehl kombinieren.

- Er kann zudem noch Feed-back-Information angeben, die bis zu 48 Zeichen umfassen darf.

Der Befehl EDIT ermöglicht es in PUMA, Skalare gemeinsam mit alphanumerischen Zeichenketten ähnlich wie in PLANIT mit print auszugeben.

Dem align von PLANIT entspricht in PUMA das VALUE. Es erlaubt eine beliebige Plazierung eines Variablenwertes oder eines Textstrings in mehreren Ausgabeformaten.

Die Unterstützung der Tabellen- und Graphikausgabe ist mit PLANIT vergleichbar.

Der Lernende kann in PUMA nach seiner eigenen Wahl ein Fernschreib- oder ein alphanumerisches Bildschirmterminal verwenden. Der Zusammenschluß mehrerer Adressatenplätze ist auch in PUMA nicht möglich.

2.3.3.2.3. TS

Als einziges Ausgabemedium ist im TS ein alphanumerischer Bildschirm mit 1000 Zeichen vorgesehen.

Die Ausgabe von Textkonserven erfolgt mit einem Unterprogrammaufruf, in dem die Nummer der gewünschten Textkonserve auf dem Textfile als Parameter übergeben wird.

Variable numerische Werte können beliebig in die Texte eingelagert werden. Die Ausgabe von Vektoren wird durch eine Tabellenroutine unterstützt, mit der ein Standardtyp dieser Ausgabeart relativ leicht definiert werden kann.

Auch für die graphische Ausgabe stellt das TS ein Unterprogramm zur Verfügung, das bis zu drei Funktionen gleichzeitig darstellen kann. Dieses erfüllt die von uns beschriebenen Forderungen nahezu vollständig.

2.3.3.2.4. Geplanter Ausbau des TS

Als eine der nächsten Verbesserungen des TS soll die Ausgabe über Fernschreiber ermöglicht werden, um somit auch größere Datenmengen ausgeben zu können, als dies der Bildschirm erlaubt.

Zur leichteren Ausgabe von einzelnen variablen Daten und für noch nicht realisierte Tabellentypen sollen weitere Ausgaberroutinen erstellt werden.

Wenn die entsprechende Hardware beschafft werden kann, ist die Erstellung der geforderten Ausgabemöglichkeiten auch für den vollgraphischen Bildschirm vorgesehen.

2.3.4. Antwortverarbeitung

2.3.4.1. Aufgaben

Der Antwortverarbeitung kommt im Rahmen des CET nicht jene hohe Bedeutung zu, die für den Ablauf von RPU charakteristisch ist, da in den vorgeschlagenen Strategien wenig verbaler Dialog stattfindet, der hohe Anforderungen an die Antwortvergleichsroutinen setzt. Es ist vielmehr notwendig, für das CET typische Antworten leicht analysieren zu können. Sollten die hier geforderten Möglichkeiten für die CET-Programmierung bei neuen Anwendungsfällen nicht ausreichen, so müssen Antwortvergleichsroutinen vom Autor in der Sprache der Struktur oder der Rechenroutinen dazu-"gestrickt" werden können. Dafür ist es notwendig, daß der Autor auf die Antwort zeichenweise zugreifen kann¹⁾. Die Anforderungen an die Antwortverarbeitung werden daher im Vergleich zu den derzeit bereits realisierten RPU-Systemen eher zu niedrig erscheinen, sind aber nach unseren bisherigen Erfahrungen mit dem CET ausreichend.

1) Vgl. Forderung nach Character-Variablen in Punkt 2.3.1.1.

Folgende Eingabearten müssen analysiert werden können:

2.3.4.1.1. Numerische Ausdrücke

Im CET ist es oft nötig, mehrere numerische Werte formatfrei eingeben zu können. In einem sehr frei gestalteten Trainingsdialog wird es erforderlich, Variablen durch Angabe ihres Namens Werte zuweisen zu können (z.B.: $A=3$, $B=0.4$, $ALPHA=0,1$), ohne daß der Autor bei jeder Eingabe alle möglichen Variablen abfragt. Die Antwortvergleichsroutinen wie Toleranzvergleich und Exaktvergleich müssen nicht durch eigene Befehle bereitgestellt werden, da der Autor diese in der Sprache des Strukturprogrammes leicht erstellen kann, falls sie benötigt werden.

2.3.4.1.2. Algebraische Ausdrücke

Algebraische Ausdrücke wie $X=C*Y+B**2$ müssen eingegeben und berechnet werden können. Dies ist notwendig, wenn z.B. eine Formel in einem Test abgefragt wird oder wenn der Student eine Funktion eingeben soll, die für ein mathematisches Verfahren benötigt wird.

2.3.4.1.3. Auswahlantworten

Auswahlantworten, bei denen der Student eine oder mehrere Alternativen angeben kann, sind eine häufige Eingabeart des CET. Es ist somit eine besonders einfache Abarbeitung notwendig. Das System soll die Eingabe automatisch auf ihre formale Richtigkeit prüfen und bei Fehlern den Studenten zu einer neuen Eingabe auffordern. Dies muß zwar auch bei den anderen Eingabearten geschehen, ist aber bei der Auswahlantwort besonders wichtig.

2.3.4.1.4. Verbale Ausdrücke

Für das CET sind folgende Routinen zur Analyse von verbalen Eingaben ausreichend:

- Exaktvergleich mit und ohne Berücksichtigung von Leerstellen

Muß der Lerner eine Zeichenkette eingeben, die genau mit der vom Autor vorgesehenen übereinstimmt, so soll ein Vergleich Zeichen für Zeichen durchgeführt werden.

- Prozentualvergleich

Gewöhnlich ist für die Eindeutigkeit der Antwortanalyse kein Exaktvergleich notwendig; orthographische Fehler oder eine vom Autor nicht vorgesehene grammatikalische Form sollen meist nicht zur Zurückweisung einer Antwort führen.

Eine Analyse der falschen Antworten anhand eines Antwortprotokolls eines Lehrprogrammes im CET hat gezeigt, daß bei einem Prozentualvergleich (mit ca. 70 %) keine tatsächlich falsche Antwort akzeptiert und keine inhaltlich richtige abgelehnt worden wäre. Der Prozentualvergleich bringt keine schlechteren Ergebnisse als der phonetische, hat jedoch wesentlich niedrigere Verarbeitungskosten.

- Schlüsselwort- und Wortteil-Vergleich

Im Rahmen des CET kann es vor allem für das Selbsttestsystem und das Lexikon notwendig sein, aus einer Antwort ein bestimmtes Wort herauszusuchen. Wird nur ein Wortstamm benötigt, soll auch dieser von der Vergleichsroutine gesucht werden können. Es soll dabei die Nummer des Zeichens erhoben werden, ab dem das gesuchte Wort geschrieben ist, um im Bedarfsfalle auch eine Reihenfolgenanalyse zu ermöglichen. Der Prozentualvergleich muß mit dem Schlüsselwort- und Wortteil-Vergleich kombinierbar sein.

2.3.4.1.5. Befehle

Zur Steuerung des Dialogablaufes steht dem Studenten eine Menge von Befehlen zur Verfügung, von denen er auch mehrere gleichzeitig eingeben können muß. Der CET-Autor soll

aber mit deren Analyse nicht belastet werden. Diese muß bei der Antwortanalyse automatisch nach jeder Eingabe durchgeführt werden. Bei Entdecken eines Kommandos soll aus der normalen Antwortverarbeitung in eine spezielle Befehlsverarbeitungsroutine gesprungen werden.

2.3.4.2. Realisierung in vorhandenen Systemen

2.3.4.2.1. PLANIT

Der Autor kann die Antworten nur innerhalb des question-frames analysieren. Im multiple choice frame wird die Studenteneingabe automatisch abgearbeitet, während im decision und programming frame keine Eingabe möglich ist.

- numerische Ausdrücke

Zur Eingabe von numerischen Werten ist in PLANIT die Variable RESPONSE vorgesehen. Dies bedeutet, daß pro Ein-/Ausgabezyklus genau eine Zahl eingegeben werden kann.

Diese Restriktion kann durch die Definition einer function umgangen werden. Der Adressat kann diese im Calc-Mode aufrufen und seine Daten als Übergabeparameter eingeben. Dieser Weg ist aber für den Autor wie für den Studenten wenig komfortabel und erlaubt nur die Eingabe einer festen Zahl von Werten. Eine numerische Eingabe bei gleichzeitiger Zuweisung zu einer Variablen durch den Studenten (z.B. A=3) ist mit dem Prozessor recycle möglich, allerdings etwas aufwendig zu programmieren. Diese Art der Eingabe ist aber auch über den Calc-Mode realisierbar.

- algebraische Ausdrücke

Mit Hilfe der Antwortvergleichsroutine FORMULAS wird eine numerische Überprüfung eines vom Studenten eingegebenen algebraischen Ausdrucks möglich. Dieser kann jedoch nicht für interne Berechnungen benutzt werden.

- Auswahlantwort

Auswahlantworten werden im multiple choice frame automatisch abgearbeitet, doch ist nur die Eingabe einer einzigen Alternative möglich.

- verbale Ausdrücke

Zur Analyse von verbalen Ausdrücken werden in PLANIT mehrere Routinen zur Verfügung gestellt. Allerdings fehlt der prozentuale Vergleich. Der phonetic match ist auf die englische Sprache bezogen und bietet kaum eine Hilfe. Es ist aber möglich, Wortstämme zu analysieren.

- Die Analyse von Befehlen mit und ohne Parameterübergabe ist sehr einfach. Der Autor kann functions definieren und dem Studenten zur Verfügung stellen.

Beispiel:

Die Studenteneingabe "<Beginn" ruft die function "BEGINN", die in einem decision-frame folgendermaßen definiert sein könnte:

```
1  3.50 d
2  function beginn = goto 1  c: allow beginn
```

Mit diesem Befehl könnte der Student von jedem Punkt des Programmes auf den Beginn zurückspringen, sofern der Befehl jeweils durch allow zugelassen ist (vgl. Teil 3, Punkt 2.1.2., frame 1.3).

2.3.4.2.2. PUMA

Da PUMA keine frame-Struktur wie PLANIT besitzt, können die Antworten an beliebigen Stellen innerhalb eines Knotens analysiert werden. Sie können auch bei Übereinstimmung mit einer vorgegebenen Antwort noch weiter untersucht und somit auf mehrere Inhalte geprüft werden.

- numerische Ausdrücke

Eine vom Studenten von Fall zu Fall festzulegende Anzahl von Variablen kann mit oder ohne Text eingegeben werden. Die Zuweisung von Variablennamen und Werten ist

ebenfalls aufgrund der Antwortanalyse möglich, wenn auch etwas umständlich zu programmieren.

- algebraische Ausdrücke

Die Prüfung algebraischer Ausdrücke ist in der Antwortanalyse nicht vorgesehen, könnte aber mit großem Programmieraufwand in der Autorensprache als Interpreter geschrieben werden.

- Auswahlantworten

Die automatische Analyse von Auswahlantworten ist in PUMA nicht vorgesehen. Auch diese muß vom Autor programmiert werden.

- verbale Ausdrücke

In PUMA ist weder ein prozentualer noch ein phonetischer Vergleich vorhanden. Es kann lediglich ein Exaktvergleich mit oder ohne Berücksichtigung von Leerstellen durchgeführt werden. Dagegen ist in PUMA ein Schlüsselwortvergleich möglich, bei dem die Nummer jenes Zeichens angegeben wird, ab dem das gesuchte Wort beginnt.

- Befehle

Eine automatische Untersuchung der Eingabe auf Befehle ist nicht vorgesehen.

2.3.4.2.3. TS

Die Eingabe einer beliebigen Anzahl von Daten ist im TS möglich. Variablennamen können - allerdings mit hohem Programmieraufwand - gemeinsam mit Variablenwerten eingegeben werden.

Die Analyse algebraischer Ausdrücke ist nicht vorgesehen.

Auswahlantworten werden vom TS auf Plausibilität überprüft; ihr Ergebnis wird an die Struktur übergeben.

Die Analyse verbaler Ausdrücke ist wegen ihrer relativ geringen Bedeutung für das Entscheidungstraining noch wenig ausgebaut. Es ist ein Exaktvergleich mit und ohne Berücksichtigung von Leerstellen mit einigem Programmieraufwand möglich, doch fehlen Prozentualvergleich und Schlüsselwortsuche.

Eine Möglichkeit zum Einbau neuer Befehle ist vorgesehen.

Da die Antworten im TS dem Autor zur Verfügung stehen, kann er diese durch eigene Routinen beliebig analysieren.

2.3.4.2.4. Geplanter Ausbau des TS

Folgende Routinen zur Unterstützung der Antwortanalyse sollen im TS als nächstes geschaffen werden:

- Zuweisung von Variablenwerten zu Variablennamen,
- Einfacherer Exaktvergleich,
- Prozentualvergleich,
- Schlüsselwort- und Wortteil-Vergleich und
- Befehlsanalyse.

Die Analyse und Interpretation von algebraischen Ausdrücken ist für eine spätere Ausbauphase vorgesehen.

2.3.5. Befehlsinterpretierer

2.3.5.1. Aufgaben

Zur Steuerung des Dialoges kann man

- Auswahlantworten und/oder
- Befehle

verwenden¹⁾. Werden nur Auswahlantworten eingesetzt, so

1) Vgl. dazu das Beispiel in Teil 2, Punkt 1.

muß der Lernende ständig eine große Zahl von Fragen beantworten. Setzt man dagegen nur Befehle ein, so benötigt der Adressat eine lange Anlernzeit, um mit dem CET arbeiten zu können. Vermutlich liegt eine vernünftige Lösung in der Kombination beider Möglichkeiten. Für Alternativen, die sich dem Lernenden immer wieder stellen, sollten Befehle vorgesehen werden, während für Handlungsmöglichkeiten, die nur sehr selten vorkommen können, der Auswahlantwort der Vorzug zu geben sein dürfte.

Die hier beschriebenen Befehle müssen als fester Bestandteil des CET-Systems in allen Strategien verfügbar sein. Sollten bei der Erstellung weiterer CET-Teachware zusätzliche Befehle notwendig werden, müssen diese leicht eingebaut werden können.

Im folgenden sind Befehle aufgezählt, die nach den Erfahrungen mit einem Versuchssystem notwendig erscheinen; sie dienen der Steuerung des Lerndialoges, z.B. Auswahl der Strategie, dem Aufruf von Dienstleistungen des Systems und der Modifikation von Parametern.

2.3.5.1.1. Globalsteuerung

Der Lernende muß die Möglichkeit besitzen, sich im CET möglichst frei zu bewegen, und somit - unterstützt von der Lernersteuerung - jene Lernstrategien und Lerninhalte wählen können, die seinen individuellen Lerneigenschaften am besten entsprechen. Dazu müssen Befehle zum Aufruf folgender Lernstrategien bereitgestellt werden:

- Selbsttests,
- Informationssystem,
- Programmierter Unterricht,
- Training,
- Problemlösung und
- Lernersteuerung.

Um sich innerhalb des Lehrangebotes zurechtfinden zu können, muß der Student auch einen Befehl haben, mit dem er sich eine Übersicht über den im System vorhandenen Lehrstoff ausgeben lassen kann.

Ist der Student aus einer Strategie ausgesprungen, ohne den begonnen Kursteil zu beenden, muß er wieder an den Absprungpunkt zurückkehren können. Ein Sprung in eine Dialogstruktur, in der er Daten manipulieren kann, muß ihm von allen Punkten des Dialoges durch einen Befehl ermöglicht werden.

Zur Beendigung des Unterrichtsdialoges muß ebenfalls ein Befehl vorhanden sein.

2.3.5.1.2. Dialogablaufsteuerung

Auch innerhalb einer Strategie muß der Lernende die Möglichkeit besitzen, den Dialog zu steuern. Dazu sind folgende Befehle wünschenswert:

- Merken des Startpunktes eines Teildialoges
Stehen dem Lernenden an einem Punkt des Dialoges mehrere Möglichkeiten offen, den Dialog fortzusetzen, so soll er diesen Entscheidungspunkt speichern können, um dort wieder aufzusetzen, wenn er im Laufe des Dialoges (z.B. im Informationssystem oder in einer Problemlösung) bemerkt, daß er eine falsche "Spur" verfolgt hat.
- Rücksprung auf Startpunkt eines Teildialoges
Um einen derartigen Teildialog abbrechen und auf den Entscheidungspunkt zurückspringen zu können, muß ebenfalls ein Befehl vorhanden sein.
- Merken einer Bilderfolge
In einem Lerndialog kann es sehr nützlich sein, bei Bedarf die letzten Bilder noch einmal "durchblättern" zu können. Dazu ist es notwendig, durch einen Befehl einen Punkt im Dialog angeben zu können, ab dem die Bilder gemerkt werden.

- Merken eines einzelnen Bildes

Ein Bild, das im späteren Dialogablauf noch einmal benötigt wird, soll gespeichert werden können, ohne daß auch die darauffolgenden Bilder gemerkt werden.

- Abrufen von gemerkten Bildern

Bei Bedarf muß der Lernende das oder die gemerkten Bilder wieder abrufen können.

- Löschen von gemerkten Bildern

Benötigt der Student die gespeicherten Bilder nicht mehr, muß er sie löschen können, um den Speicheraufwand zu verringern.

- Sprung auf Verteiler

Hat sich der Student für einen bestimmten Lernstoff innerhalb einer Strategie entschieden, kann es für ihn nützlich sein, bereits nach Durcharbeitung eines Teiles dieses Lernstoffes noch andere Lerninhalte im Rahmen dieser Strategie durchzuarbeiten. Dazu muß er mit Hilfe eines Befehles auf einen Verteiler springen können, von dem aus er andere Lerninhalte wählen kann.

- Datenmanipulation

Im Rahmen der Datenbereitstellung muß der Student Befehle zum Speichern, Löschen, Generieren und Sortieren haben.

2.3.5.1.3. Aufruf von Systemdienstleistungen

Folgende Befehle dienen der Erhöhung des Dialogkomforts:

- Erstellung von Hardcopies

Zur Unterstützung des Dialoges und des Studiums zu Hause ist es oft wichtig, Unterlagen zu haben, die bei Benutzung eines Bildschirmterminals nicht automatisch anfallen. Der Lernende soll die Möglichkeit haben, diese Unterlagen

- mit Hilfe eines Fernschreibers oder
 - eines Schnelldruckers
- erstellen zu lassen.

- Kontakt zum Lehrer und zur Systembetreuung
Der Student soll Fragen an den Lehrer eingeben können, die dieser periodisch beantwortet.
- Ausgabe
Der Lernende soll auch außerhalb der fest vorgesehenen Tabellen und Graphiken selbst die Werte von Variablen ausgeben lassen können. Er soll die Möglichkeit besitzen, nach einem Aufruf einer Tabellen- oder Graphikfunktion seinen Ausgabewunsch zu spezifizieren.
- Befehlsliste
Ein Befehl soll es dem Lernenden ermöglichen, sich eine Liste der Befehle mit Erklärungen ausgeben zu lassen.

2.3.5.1.4. Modifikation von Parametern

Zur raschen Änderung von Parametern eines Verfahrens ist es nützlich, diese durch willkürliche Eingabe des Studenten wie folgt modifizieren zu können:

*A=30.1 *B=0.3 *ALPHA=0.1

Diese Art der Parameteränderung ist der durch Auswahlantworten vom Strukturprogramm gesteuerten dann überlegen, wenn der Lernende ein Verfahren oft verwendet und daher nicht immer die sonst notwendigen mühsamen Fragen beantworten will.

2.3.5.2. Realisierung in vorhandenen Systemen

Da die vorhandenen Systeme primär zur Verwirklichung von rechnergesteuerten (im Unterschied zum lernergesteuerten) Unterricht geschaffen wurden, sehen sie wenig Möglichkeiten für die Bewegungsfreiheit des Studenten vor. Der Autor kann zwar einige Befehle der hier geforderten durch hohen Programmieraufwand dem Studenten bereitstellen. Es ist aber in den Systemen PLANIT und PUMA nicht möglich, diese

Programmteile als Standardunterprogramme zu definieren, da die Beschränkungen in der Unterprogrammtechnik dies nicht erlauben.

2.3.5.2.1. PLANIT

In PLANIT stehen dem Studenten folgende Befehle zur Verfügung:

- DIAL: Der Student kann mit anderen Adressaten kommunizieren.
- <(oder CALC): Die Tischrechnerfunktion CALC wird aufgerufen.
- >(oder READY): Beendigung des Tischrechnermodus.
- HELP: PLANIT liefert eine Kurzbeschreibung einer gewünschten Funktion, einer Matrix oder eines Items.
- REVIEW: Der Student kann auf vom Autor gekennzeichnete frames springen und Wiederholungssequenzen durcharbeiten.
- ?: Der Benutzer erhält eine Kurzbeschreibung der letzten Meldung von PLANIT.
- GET: Damit wird eine Terminalsitzung eröffnet und ein Lehrprogramm aufgerufen.
- <FINISHED und OUT: Dienen zur Beendigung der Terminalsitzung.

Damit wird nur ein sehr kleiner Teil der oben beschriebenen Befehle abgedeckt. Der Großteil davon könnte durch teils sehr einfache, teils etwas schwierigere Programmierung durch den Autor erstellt werden.

Folgende Befehle können in PLANIT überhaupt nicht oder nur sehr schwer realisiert werden:

- Aufruf der Lernersteuerung, da diese, wie sie hier verstanden und im Teil 2, Punkt 4, beschrieben wird, in PLANIT nicht realisierbar ist.
- Speichern und Löschen von Daten auf Externspeicher, da von PLANIT aus kein File auf einem Externspeicher angesprochen werden kann.

- Merken des Startpunktes eines Teildialoges und Rücksprung auf diesen, weil der aktuelle Stand des Programmes nicht auf einen Externspeicher gelegt werden kann.
- Merken, Abrufen und Löschen von Bildern, da Texte nicht unabhängig vom Ausgabebefehl gespeichert werden können.

2.3.5.2.2. PUMA

PUMA bietet dem Studenten folgende Befehle an:

- PU: Zur Eröffnung und
- OFF: zur Beendigung der Sitzung.
- WIEDERHOLUNG: Der Student kann ein Wiederholungsprogramm aufrufen.
- LEXIKON: Zur Klärung einzelner Begriffe steht ein Lexikon zur Verfügung.
- STATUS: Mit STATUS wird der aktuelle Stand der lernerindividuellen Variablen abgefragt.
- LEHRER: Mit dem Kommando Lehrer erhält der Student die Möglichkeit, folgende Befehle einzusetzen:
 - Wertzuweisung: Der Lernende kann beispielsweise mit dem Befehl X=3 der Variablen X den Wert 3 zuweisen.
 - KNOTEN: nn: Mit diesem Befehl ist der Sprung an beliebige Punkte im Programm möglich.
 - ENDE: Mit ENDE kehrt der Student in den normalen Programmablauf zurück.

Auch in PUMA ist somit nur ein sehr kleiner Teil der geforderten Befehle vorhanden.

Das Erstellen neuer Befehle durch den Autor gestaltet sich aber wesentlich schwieriger als in PLANIT, da in PUMA bei jeder Eingabe auf das Vorhandensein eines Befehles abgefragt werden müßte. Die Programmierung folgender Befehle

durch den Autor ist mit zu großem Aufwand verbunden oder nicht möglich:

- Merken des Startpunktes eines Teildialoges und Rücksprung auf diesen,
- Merken und Abrufen von Bildern,
- Erstellung von Hardcopies und
- Ausgabe von Tabellen und Graphiken.

2.3.5.2.3. TS

Im TS wurden bisher drei Befehle realisiert:

- BEGINN: Der Student kann von einem beliebigen Punkt des Programmes auf den Beginn der Trainingslektion springen.
- DATEN: Mit diesem Kommando verzweigt er zur Datenbereitstellung.
- OFF: Der Befehl wird zur Beendigung des Trainings benutzt.

2.3.5.2.4. Geplanter Ausbau des TS

Sämtliche geforderten Befehle sollen im TS verwirklicht werden.

2.3.6. Datenbereitstellung

2.3.6.1. Aufgaben

Besonders für die Trainingsstrategie ist es wichtig, Datenfiles auf einem Externspeicher wie Platte oder Trommel anlegen zu können, um größere Datenbereiche, die für betriebswirtschaftliche Verfahren kennzeichnend sind, nicht internspeicherresident halten zu müssen. Das CET-System soll den Zugriff auf beliebige - auch nicht vom CET aus eröffnete - Files erlauben und damit auch an Datenbanksysteme anschließbar sein. Die Datenbereitstellung muß sowohl dem Autor für

die Erstellung von CET-Kursen als auch dem Lernenden bei der Problemlösung zur Verfügung stehen. Sie soll folgende Möglichkeiten bieten:

- Dateneingabe

Beliebige Daten sollen in logische Sätze gegliedert über Terminal oder Lochkarte bzw. andere Datenträger eingegeben werden können.

- Datengenerierung

Das Unterrichtssystem soll die Generierung von Daten für Simulationen und zur Erstellung von schülerindividuellen Problemstellungen unterstützen. Dazu soll der Benutzer Angaben über folgende Eigenschaften der Daten machen können:

- die Funktion, um welche die Werte schwanken sollen,
- den Bereich, innerhalb dessen ein zu bestimmender Prozentsatz der Daten liegen soll, und
- die Verteilung, der die Werte gehorchen sollen.

Funktionen: Als Funktionen sollen beliebige vom Benutzer einzugebende algebraische Ausdrücke angegeben werden können. Beispiel: $A + B * X + C/2 * X ** 2$

Bereich: Zur Definition des Bereiches soll die erlaubte Schwankung um die Funktion absolut oder prozentual angegeben werden können. Dazu ist noch die Angabe des Anteils der generierten Werte, die in diesem Bereich liegen, in Prozenten notwendig.

Beispiel:

90% der Daten sollen innerhalb des Bereiches +10% und -10% liegen.

Verteilungen: Es müssen mindestens folgende Verteilungen wählbar sein:

- Rechteck-,
- Normal- und
- Poissonverteilung.

Weitere Verteilungen müssen bei Bedarf leicht eingebaut werden können.

Die Angaben über die Art der Datengenerierung soll das System vom Studenten On-line abfragen. Der Autor muß dazu einen Off-line-Aufruf besitzen.

- Sortieren

Zum Sortieren der Sätze sollen dem Benutzer Sortierrou-tinen zur Verfügung gestellt werden.

- Datensicherung

Die Daten sollen auf den Zugriff von

- Einzelpersonen,
- Personengruppen und
- allen (keine Beschränkung)

begrenzt werden können.

Die Files sollen periodisch auf Band kopiert werden, um die Folgen einer zufälligen Zerstörung einer Plattendatei gering zu halten.

- Dateiübersicht

Auf Wunsch des Benutzers soll eine Übersicht über die von ihm eröffneten und die ihm zugänglichen Bestände ausge-geben werden, in der folgende Daten enthalten sein sollen:

- Namen der Files,
- Fileübersicht,
- Anzahl der Sätze,
- Satzlänge,
- Datum der letzten Datenänderung und
- vom Benutzer fakultativ einzugebender Kommentar
(z.B.: Satzaufbau, Sortierkriterien).

2.3.6.2. Realisierung in vorhandenen Systemen

RPU-Systeme sind nicht für die Manipulation von größeren Datenmengen konzipiert, da dies für die Darbietung von Pro-

grammiertem Unterricht nicht notwendig ist.

2.3.6.2.1. PLANIT

In PLANIT kann weder der Autor noch der Lernende Datenfiles auf Platte oder Band anlegen. Datenbereiche müssen daher im Programm gehalten werden und sind somit in ihrer Größe (Installationsparameter) stark begrenzt.

PLANIT weist folgende Mängel bei der Erfüllung unserer Anforderungen auf:

- Dateneingabe ist nur on-line möglich.
- Die Generierung von Daten wird nur durch die Erzeugung von rechteck- und normalverteilten Zufallszahlen unterstützt.
- Die Datenänderung kann nur on-line erfolgen.
- Sortierroutinen sind nicht vorhanden.
- Daten sind nur dem Autor und jeweils einem Studenten zugänglich. Sie werden mit den übrigen Lehrprogrammdateien gesichert.
- Eine Übersicht über vorhandene Dateien ist nicht vorgeesehen.

2.3.6.2.2. PUMA

Als Besonderheit für eine RPU-Sprache erlaubt PUMA den Zugriff auf Plattenfiles, die auch off-line erstellt werden können. Von den oben aufgestellten Anforderungen an die Datenbereitstellung werden folgende nicht erfüllt:

- Es ist kein Zufallszahlengenerator vorhanden.
- Ein Zugriff des Studenten auf die Daten muß vom Autor programmiert werden.
- PUMA stellt keine Sortierroutinen zur Verfügung.
- Auch eine Dateiübersicht müßte vom Autor erstellt werden.

2.3.6.2.3. TS

Durch die Benutzung von FORTRAN ist es im TS möglich, auf beliebige Files auf Externspeichern zuzugreifen. Der Anschluß einer Datenbank ist jedoch noch nicht realisiert.

Der Student hat keinen Zugriff auf Plattenfiles. Dadurch ist die Anzahl der Daten, die über Terminal eingegeben werden können, auf die im Programm reservierten Bereiche (40 Werte) beschränkt. Datensicherungsroutinen und eine Dateiübersicht sind daher ebenfalls nicht vorhanden.

Die Datengenerierung wird nur durch eine Routine zur Erzeugung rechteckverteilter Zufallszahlen unterstützt.

2.3.6.2.4. Geplanter Ausbau des TS

Beim Ausbau des TS sollen die Möglichkeiten der Datenbereitstellung stark erweitert werden. In der ersten Stufe soll auch dem Studenten der Zugriff auf Plattenfiles ermöglicht und das dazu benötigte Service wie Datensicherung und Dateiübersicht angeboten werden. Ebenso sollen die geforderten Routinen zur Datengenerierung geschaffen werden.

Für eine spätere Phase ist auch der Anschluß eines Datenbanksystems geplant.

2.3.7. Lerndatenaufzeichnung und -auswertung

2.3.7.1. Aufgaben

Auf Wunsch des Autors sollen vom CET-System folgende Daten während des Lerndialoges schülerindividuell erhoben werden:

- Weg durch die Kurse und Lektionen,
- Eingabe des Studenten,
- Richtigkeit der Studentenantwort,
- Antwortzeiten,
- Rechenzeit und
- Blockierungszeit des Terminals.

Die Daten, die beim Testen einer CET-Lektion durch den Autor entstehen, sollen auf einen eigenen File gebracht werden, da sie sonst die Mittelwerte und Streuungen bei der Versuchspopulation verzerren.

Die Lerndaten sollen dem Autor zur Kursrevision und dem Unterrichtsforscher zur Analyse des Lernprozesses zur Verfügung stehen. Dazu muß ein geeignetes Auswertungsprogramm vorhanden sein.

Bei den derzeit verwendeten Dialogstatistiken kapituliert der Autor oft vor der Datenmenge, die er kaum auswerten kann, da die Daten meist nicht bedarfsgerecht aufbereitet sind.

Der Autor und der Unterrichtswissenschaftler wollen anhand der Kursdaten Probleme wie die folgenden lösen:

- Wie hoch ist die durchschnittliche Bearbeitungszeit eines Bildes?
Welche Bilder haben die stärksten Abweichungen von diesem Wert?
Wie kommen die Unterschiede zustande?
- Welche Programmwege werden selten oder nie benutzt?
- Sind nur richtige Antworten als richtig und nur falsche als falsch erkannt worden?
- Welche falschen Antworten gab es und wo traten sie auf?

Sollen solche Fragen anhand eines einzigen Statistik-Out-puts gelöst werden, so erfordert dies einen nicht vertretbaren Zeitaufwand, vor dem man meist zurückschreckt. Dem Autor muß somit die Möglichkeit geboten werden, Ausnahmefälle zu definieren und sich diese ausgeben zu lassen. Dazu sollten die beiden unten behandelten Tabellen-Typen abrufbar sein, für die der Autor Auswertungskriterien angeben können soll. Möchte der Autor beispielsweise Bilder

ermitteln, bei denen der Student Schwierigkeiten hatte, könnte er folgende Ausgabespezifikation angeben:

Beispiel: Kursstatistik für die Lektion "Prognose",
wenn im Wort 127 des Satzes ein "-" steht.

Er bekäme damit für eine bestimmte Lektion sämtliche falschen Antworten aufgelistet.

Für den Autor sind vor allem folgende zwei Tabellentypen interessant:

(1) Studentenstatistik

Sie soll geordnet sein nach:

- Student
- Lektion
- Bild

Folgende Informationen sollen für die verschiedenen Verdichtungsstufen ausgegeben werden:

Tabellenkopf:

- Kurs
- Lektionen
- durchschnittliche Bearbeitungszeit der Lektionen
- Streuung der Bearbeitungszeiten
- durchschnittliche Antwortzeit
- Streuung der Antwortzeiten
- durchschnittliche Rechenzeit
- Streuung der Rechenzeit
- durchschnittliche Anzahl der Bilder pro Lektion
- durchschnittliche Anzahl der Bearbeitungen der Bilder (Wie oft wurde dasselbe Bild gezeigt?)

Kopf pro Student:

- Lektionen (bearbeitete)
- durchschnittliche Bearbeitungszeit der Lektionen
- durchschnittliche Antwortzeit
- Streuung der Antwortzeiten

- Gesamtrechnenzeit des Studenten
- durchschnittliche Anzahl der Bilder pro Lektion
- durchschnittliche Anzahl der Bearbeitungen der Bilder

Kopf pro Lektion:

- Bearbeitungszeit
- Rechenzeit
- durchschnittliche Antwortzeit
- Streuung der Antwortzeiten
- Anzahl der Bilder
- durchschnittliche Anzahl der Bearbeitungen der Bilder

Bild: (in der Reihenfolge des Dialogablaufes)

- Nummer des Bildes (Wieviertes Bild des Studenten innerhalb der Lektion?)
- Bildart (entspricht ungefähr der Frame-Nummer)
- Nummer der Bearbeitung (Wievierte Bearbeitung dieses Bildes?)
- Antwort
- richtig/falsch
- Verzweigung in anderes Programm

(2) Kursstatistik

Sie soll nach folgender Hierarchie aufgebaut sein:

- Kurs
 - Lektion
 - Bild
 - Student
 - Durcharbeitung

Die Daten für die einzelnen Stufen der Hierarchie sollen analog zur Studentenstatistik gebildet werden.

Neben den hier beschriebenen Standardtabellen, die durch Kriterienangabe vom Autor modifiziert werden können, soll ein direkter Zugriff auf die Dialogstatistik oder eine Kopie mit eigenen Auswertungsprogrammen möglich sein.

Dem Student soll auf Wunsch ein vollständiges Protokoll seiner Terminalsitzung auf dem Schnelldrucker ausgegeben werden, damit er wesentliche Teile daraus off-line noch einmal durcharbeiten kann.

2.3.7.2. Realisierung in vorhandenen Systemen

Der Kursrevision und somit der Lerndialogstatistik werden im RUU bisher wenig Aufmerksamkeit gewidmet. Dies liegt einerseits daran, daß die Pädagogik wenig Aussagen darüber macht, wie ein optimal wirksames Lernprogramm aufgebaut sein soll, andererseits aber an der Tatsache, daß die meisten derzeit erstellten Unterrichtsprogramme nur für einen sehr kleinen Adressatenkreis erstellt werden und daher die hohen Fixkosten für umfangreiche Programmverbesserungen bei den Kosten pro Ausbildung und Student zu hoch ins Gewicht fallen würden. Dazu kommt noch, daß die meisten RUU-Systeme einen Statistik-Output liefern, der als "Zahlenfriedhof" zu bezeichnen ist und den Autor von der Verwendung abhält.

2.3.7.2.1. PLANIT

PLANIT erlaubt grundsätzlich die Ausgabe folgender Tabellen:

- history und
- history + summary.

In der history werden jene frames ausgegeben, in denen eine Eingabe erfolgt. Wird im Lernprogramm die Funktion TRACE gesetzt, werden alle frames aufgelistet.

Im summary werden einige zusammenfassende Daten dargestellt. Der Autor hat zudem die Möglichkeit, die Statistik wahlweise

für Programmteile oder das gesamte Programm ausgeben zu lassen.

Die oben an eine Lerndialogstatistik gestellten Anforderungen werden in PLANIT nur mangelhaft erfüllt:

- Es werden keine Rechenzeiten erhoben.
- Es können stets nur die Daten eines einzelnen Schülers aufgelistet werden. Somit fehlen auch Übersichtsdaten über eine größere Schülergruppe.
- Es können nur sehr wenige Kriterien für die Auswertung (nur Framenummern) bestimmt werden.

2.3.7.2.2. PUMA

PUMA liefert eine Statistik, in der für jeden Lernprogrammabschnitt Daten wie die durchschnittliche Bearbeitungszeit ausgegeben werden. Folgende der oben beschriebenen Anforderungen erfüllt PUMA nicht:

- Es zeichnet nicht auf, ob die Antworten richtig oder falsch waren.
- Rechenzeiten werden nicht registriert.
- Der Autor kann keine Kriterien für die Auswertung definieren.
- Die Auswertung ist in PUMA nur lehrstoffbezogen.
- Der Autor kann nicht mit eigenen Programmen auf die Lerndateien zugreifen.

2.3.7.2.3. TS

Im TS ist keine Lerndatenaufzeichnung vorhanden.

2.3.7.2.4. Geplanter Ausbau des TS

Die Lerndatenaufzeichnung soll sämtliche geforderten Schülerdaten erfassen und die in den Anforderungen beschriebenen Auswertungen liefern. Auch die Möglichkeit zur Angabe von Auswertungskriterien ist geplant.

2.3.8. Eingabe des Kursmaterials

2.3.8.1. Aufgaben

Zur Erhöhung der Übersichtlichkeit von Dialogprogrammen ist es notwendig,

- Texte,
 - Strukturen und
 - Rechenroutinen
- getrennt einzugeben.

- Eingabe der Texte

Die Texte machen mengenmäßig den bedeutendsten Teil von CET-Programmen aus. Bereits ein relativ kleines Lernprogramm hat ca. 1000 Textzeilen.

Die beiden folgenden Eingabemöglichkeiten erscheinen uns besonders geeignet:

(1) Eingabe über Lochkarte

Der Autor schreibt den Text ohne Berücksichtigung der Zeilengrenzen unter Verwendung von Steuerzeichen für den Zeilenwechsel, wenn er diesen an ganz bestimmten Stellen will. Andernfalls stellt das System die Zeilen zusammen. Diese Variante hat den Vorteil, daß die zeitraubende Tätigkeit der Texteingabe vom Autor an eine Locherin abgegeben werden kann.

(2) Eingabe über Bildschirm

Der Bildschirm eignet sich besonders gut zur Eingabe von Texten, da das Bild bereits bei der Eingabe in seinem Ausgabeformat betrachtet werden kann. Zudem ist eine Korrektur am Bildschirm relativ leicht durchführbar.

Die On-line-Eingabe von Texten ist besonders dann vorzuziehen, wenn diese in einem bestimmten Format (z.B. als Rahmen einer Tabelle) erscheinen müssen.

Die Durchführung von Korrekturen (Tippfehler u.ä.) ist auf dem Bildschirm wesentlich einfacher als bei der Off-line-Eingabe, da Zeichen überschrieben werden können. Auch die Texteingabe über Bildschirm kann von einer Datentypistin durchgeführt werden.

Da beide Eingabearten relativ einfach zu erstellen sind und jede für bestimmte Situationen Vorteile bietet, sind auch beide zu fordern.

In jedem Falle muß der Autor eine Liste der Texte anfordern können.

- Eingabe der Struktur

Eine umfangreiche Struktur kann nicht ohne Vorbereitung direkt "in die Maschine" programmiert werden. Der Autor muß das Programm off-line erstellen. Liegt es aber fertig vor, so kann es auch off-line, wie dies bei problemorientierten Sprachen üblich ist, eingegeben und korrigiert werden.

Folgende Listen sollten wahlweise ausgegeben werden können:

- die Struktur allein und
- die Struktur zusammen mit den Texten.

- Eingabe der Rechenroutinen

Da die Rechenroutinen in problemorientierten Sprachen geschrieben werden sollen, sind sie für den jeweiligen Compiler einzugeben.

2.3.8.2. Realisierung in vorhandenen Systemen

In den RPU-Sprachen wird die Lehrprogrammeingabe sehr oft on-line durchgeführt. Das Strukturprogramm, die Lehrtexte und die Rechenroutinen werden vom Autor in einem speziellen Modus über Terminal eingetippt. Der Autor erhält dabei sog.

promptings durch das System. Auf diese Art soll auch in der Datenverarbeitung unerfahrenen Autoren das Arbeiten mit dem Rechner ermöglicht werden.

Die meisten Systeme erlauben daneben aber auch die Off-line-Eingabe, manche sogar nur diese.

2.3.8.2.1. PLANIT

PLANIT gestattet die On-line- und Off-line-Eingabe von Lernprogrammen. Texte, Struktur und Rechenroutinen werden in PLANIT nicht getrennt.

2.3.8.2.2. PUMA

PUMA sieht die Off-line-Eingabe der Programme, getrennt in den Text und den Strukturteil, in dem auch die Rechenroutinen enthalten sind, vor. Textkorrekturen können über Bildschirm durchgeführt werden.

2.3.8.2.3. TS

Im TS werden das Strukturprogramm und die Rechenroutinen vom FORTRAN-Compiler übersetzt, während die Texte getrennt auf einen File gelesen werden. Sie können über Lochkarten oder über Bildschirm eingegeben werden.

2.3.8.2.4. Geplanter Ausbau des TS

Es soll ermöglicht werden, Texte gemeinsam mit der Struktur auslisten zu können. Die Texteingabe über Lochkarte soll in der geforderten Weise vereinfacht werden.

2.4. Zusammenfassung

Um einen zusammenfassenden Überblick über diesen Abschnitt zu schaffen, sind in Tab. 3 die Anforderungen mit ihrer Dringlichkeit für das CET bewertet und den derzeitigen Realisierungsmöglichkeiten und den Erweiterungsplänen des TS gegenübergestellt worden (Siehe Tab. 3.).

In den Zeilen der Tabelle sind die Anforderungen an ein CET-System stichwortartig aufgeführt. Es wurde jene Reihenfolge beibehalten, die wir bei der ausführlichen Beschreibung gewählt haben, um das Nachschlagen zu erleichtern. In der Spalte 1 sind zudem für die Überpunkte Hinweise auf die entsprechenden Abschnitte eingetragen.

Nach der Bezeichnung der Anforderung in Spalte 2 folgt in Spalte 3 ein Dringlichkeitscode, dem folgende Abstufung zugrunde liegt:

- 1..... wünschenswerter, aber nicht notwendiger Komfort
- 2..... notwendiger Komfort für Autor und Lernende
- 3..... Voraussetzung für die geplante Verwirklichung des CET
- 4..... Voraussetzung für die Minimalversion des CET

Anschließend an Spalte 3 werden die Realisierungsmöglichkeiten in den verschiedenen Softwaresystemen bewertet. Es wird für jedes System unterschieden, wie weit eine Anforderung realisiert ist (Spalte a) und wieviel Aufwand für den Benutzer (Autor und Schüler) entsteht (Spalte b), wobei wir folgende Codes benutzen:

Realisierung (Spalte a)

- 1..... vollständig realisiert
- 2..... zum Großteil realisiert
- 3..... ansatzweise realisiert
- 4..... nicht realisiert

Aufwand des Benutzers (Spalte b)

- 1..... sehr gering .
- 2..... mäßig
- 3..... hoch
- 4..... sehr hoch

Wenn keine sinnvolle Eintragung möglich ist, wird dies durch "-" angegeben.

Das Zeichen "x" steht, wenn in einem System eine Anforderung zwar nicht erfüllt, aber eine Ersatzlösung angeboten wird. In Spalte b wird darum vermerkt, wie aufwendig die Ersatzlösung ist.

Eine auch nur annähernde Erfüllung der Anforderungen, die von der Lehrmethode CET an ein System zu stellen sind, wird unseres Wissens in keinem der bestehenden Unterrichtssysteme erreicht.

Obwohl wir mit PLANIT und PUMA zwei Systeme ausgewählt haben, die u.E. den neuesten Stand auf dem Gebiet der Unterrichtssysteme repräsentieren, werden die Anforderungen an ein System zur Realisation von CET auch dann nicht erfüllt, wenn man eine aufwendige Programmierung durch den Autor in Kauf nimmt.

Im Projekt CET wurde daher das TS erstellt, mit dem wenigstens die Versuche mit CET durchgeführt werden konnten. Für den weiteren Ausbau des CET ist es aber notwendig, das TS um die genannten Punkte zu erweitern. Zwar kann eine Bibliothek von Hilfsroutinen, aus der das TS derzeit besteht, auf weite Sicht eine MMK-Sprache nicht ersetzen, doch dürfte der geplante Ausbau des TS wichtige Erkenntnisse für die Konzeption dieser Sprache bringen.

Vgl. Teil 4, Punkt	Anforderungen an die Systemteile	Dring- keit	PLANIT		PUMA		TS		Ausbau	
			a	b	a	b	a	b	a	b
2.3.1.	STRUKTUR									
	UP-Technik mit und ohne Parameterübergabe	4	2	2	3	2	1	1	1	1
	Definition von Makrobefehlen	1	4	4	4	4	4	4	3	2
	Drei-Weg-Verzweigungen									
	- mit festen Verzweigungen	4	x	1	1	1	1	1	1	1
	- mit variablen Verzweigungen	1	x	1	4	-	4	-	4	-
	- logische Verzweigung	2	1	2	x	2	1	1	1	1
	Sprungbefehle									
	- mit festen Sprungadressen	4	1	1	1	1	1	1	1	1
	- mit variablen Sprungadressen	1	1	1	4	-	4	-	4	-
	Berechneter Sprungbefehl	2	1	1	1	1	1	1	1	1
	Laufschleife	2	3	2	4	-	1	1	1	1
	Restartpoints									
	- automatisch	2	1	1	1	1	4	-	1	1
	- autordefiniert	1	1	1	4	-	4	-	1	1
	Definition von Statement-labels	4	2	2	1	1	1	1	1	1
	automatische Zeilennummerierung	1	3	3	1	1	1	1	1	1
	Datentypen									
	- String	3	4	-	4	-	4	-	1	1
	- Character	3	4	-	2	2	1	2	1	2
	- Real	4	1	1	1	1	1	1	1	1
	- Integer	2	4	-	1	1	1	1	1	1
	- Gleitkomma	1	4	-	4	-	1	1	1	1

Vgl. Teil 4, Punkt	Anforderungen an die Systemteile	Dring- lich- keit	PLANIT		PUMA		TS		Ausbau	
			a	b	a	b	a	b	a	b
2.3.2.	RECHENROUTINEN									
	Einbau vorhandener Rechenroutinen	2	4	-	4	-	1	1	1	1
	hohe Rechenfähigkeiten	4	1	1	1	2	1	1	1	1
	hohe Verarbeitungsgeschwindigkeit	4	4	-	1	-	1	-	1	-
	Rechenroutinen als Unterprogramme	4	3	2	3	3	1	1	1	1
2.3.3.	AUSGABEAUFBEREITUNG									
	Bestimmung des Ausgabemediums									
	- durch Hardware	3	4	-	2	-	4	-	1	-
	- durch Lernenden	3	4	-	1	-	4	-	1	1
	- durch Programmator	3	4	-	4	-	4	-	1	1
2.3.3.1.2.	- für Terminalverbund	3	4	-	4	-	4	-	1	1
	Ausgabearten									
	- Textkonserven	4	2	2	1	1	2	1	1	1
	- Text und einzelne variable Werte	4	1	2	1	2	1	3	1	1
	- Text und Tabellen	4	1	4	1	4	1	2	1	1
	- Text und Graphik	4	1	4	1	4	1	2	1	1
	- Konkatenation	3	4	-	4	-	4	-	1	1
	- Textsubstitution	3	4	-	4	-	4	-	1	1
	- Routinen für vollgraphischen Bildschirm	3	4	-	4	-	4	-	1	1
2.3.4.	ANTWORTVERARBEITUNG									
	numerische Ausdrücke	4	2	3	1	1	1	1	1	1

Vgl. Teil 4, Punkt	Anforderungen an die Systemteile	Bezug- lich- keit	PLANIT		PUMA		TS		Ausbau	
			a	b	a	b	a	b	a	b
	algebraische Ausdrücke	1	3	1	4	-	4	-	3	1
	Auswahlantworten	4	2	1	1	2	1	2	1	1
	verbale Ausdrücke									
	- Exaktvergleich	4	1	1	1	2	1	3	1	1
	- Prozentualvergleich	3	4	-	4	-	4	-	1	1
	- Schlüsselwort- und Wortteilvergleich	3	2	2	2	3	4	-	1	1
	- Befehle	3	1	2	4	4	3	3	1	1
	BEFEHLSINTERPRETER									
	Globalsteuerung	4	2	2	2	2	3	3	1	1
2.3.5.	Dialogablaufsteuerung									
	- Merken des Startpunktes	3	4	-	4	-	4	-	1	1
	- Rücksprung auf Startpunkt	3	4	-	4	-	4	-	1	1
	- Merken einer Bilderfolge	3	4	-	4	-	4	-	1	1
	- Merken eines einzelnen Bildes	3	4	-	4	-	4	-	1	1
	- Abrufen von gemerkten Bildern	3	4	-	4	-	4	-	1	1
	- Löschen von gemerkten Bildern	3	4	-	4	-	4	-	1	1
	- Sprung auf Verteiler	3	4	-	4	-	4	-	1	1
	Aufruf von Systemdienstleistungen									
	- Erstellung von Hardcopies	2	-	-	4	-	2	3	1	1
	- Kontakt mit Lehrer und Systembetreuung	2	2	1	4	-	4	-	1	1
	- Ausgabe	2	3	2	4	-	4	-	1	1
	- Befehlsliste	2	4	-	4	-	4	-	1	1

Vgl. Teil 4, Punkt	Anforderungen an die Systemteile	Dring- lich- keit	PLANIT		PUMA		TS		Ausbau	
			a	b	a	b	a	b	a	b
	- Lochkarten mit großen Formatfreiheiten	2	2	2	2	2	2	2	1	1
	- Bildschirm	2	-	-	2	2	1	1	1	1
	Differenzierung bei den Programmlisten	2	4	-	1	1	4	-	1	1

Tab. 3: Dringlichkeit und Realisierung der Systemteile

F ü n f t e r T e i l

Ausblick auf den Einsatz des CET in der Praxis

Nach der Darstellung der Konzeption und der Realisierungsprobleme soll nun ein Ausblick auf den Einsatz des CET gegeben werden.

1. Einsatz in der betriebswirtschaftlichen Ausbildung an der Universität

Das CET ist ursprünglich für die betriebswirtschaftliche Ausbildung an der Universität geschaffen worden. Am Beispiel der Ausbildung im Fach Betriebsinformatik wird nun gezeigt, wie CET-Kurse in den bestehenden Lehrbetrieb eingegliedert werden können.

1.1. Einsatz des CET für die Ausbildung im Fach Betriebsinformatik

Bei der Lehre der betrieblichen Datenverarbeitung werden am Betriebswirtschaftlichen Institut der Universität Erlangen-Nürnberg (Lehrstuhl für Betriebs- und Wirtschaftsinformatik) folgende Lehrziele verfolgt:

"Es sollen ... die Grundkenntnisse der EDV an solche Studierende vermittelt werden, die später auf der Seite der EDV-Hersteller (z.B. als Systemberater, EDV-Vertriebsfachleute) oder auf der Seite der Anwender (z.B. als Systemingenieure, Systemanalytiker, Organisatoren) in der Datenverarbeitung tätig werden wollen. ... Der Absolvent des Kurses soll in der Lage sein, Grobkonzeptionen für moderne

Anwendungssysteme der integrierten Datenverarbeitung zu entwerfen, die sowohl die Routine-Datenverarbeitung als auch Optimierungsmodelle und automatische Führungsinformation (Informations- und Planungssysteme) enthalten¹⁾. Im bestehenden Ausbildungsplan (vgl. Tab. 4) ist versucht worden, diese Lehrziele durch den Einsatz mehrerer Unterrichtsmethoden möglichst gut zu erreichen. Die Anwendung von Modellen und deren Einsatz im Entscheidungsprozeß können mit herkömmlichen Lehrmethoden jedoch nicht in ausreichendem Maße vermittelt werden²⁾. Es ist daher beabsichtigt, zur Verbesserung des Unterrichts in diesem Bereich das CET einzusetzen³⁾.

1.1.1. Lehrveranstaltung "Simulation"

In der Lehrveranstaltung "Simulation" wurde der Student bisher mit den Grundlagen der Simulationstechnik und mit betriebswirtschaftlichen Anwendungsgebieten vertraut gemacht. Der hohe Rechenaufwand dieser Modelle gestattete es aber nicht, mehr als die prinzipielle Vorgehensweise und beispielhafte Anwendungsfälle zu zeigen. In Zukunft soll der Student durch das CET die Möglichkeit erhalten, betriebswirtschaftliche Dialog-Planungsmodelle mit On-line-Simulation bei der Lösung praktischer Fälle einzusetzen. Es stehen ihm zusätzlich noch einige Modelle auf deterministischer Basis für die Problemlösung zur Verfügung.

Am Beispiel eines CET-Programmes, in dem die Break-even-Analyse mit einer Risikoanalyse kombiniert wird⁴⁾, ist anschließend der Aufbau der Lehrveranstaltung dargestellt (vgl. Abb. 8):

- 1) Mertens, P., Manual über die Ausbildung in Betriebsinformatik, Betriebswirtschaftliches Institut der Universität Erlangen-Nürnberg (Lehrstuhl für Betriebs- und Wirtschaftsinformatik), Nürnberg 1972, S. 1.
- 2) Vgl. dazu Teil 1.
- 3) Die in diesem Abschnitt beschriebenen Gedanken gehen auf G. Holub zurück, die die hier nur skizzierten Überlegungen in einer in Arbeit befindlichen Dissertation näher untersucht. Vgl. Holub, G., a.a.O.
- 4) Vgl. Rackelmann, G., a.a.O.

Lehrveranstaltung	Bemerkungen zur Form
Einführung in die Hardware	Programmierte Einführung durch: Wolters, M.F., Der Schlüssel zum Computer, Düsseldorf 1970. Vertiefung ausgewählter Kapitel durch Gastvorträge, soweit möglich Demonstration von Geräten, Filme, Exkursionen.
COBOL-Kurs	Programmierte Einführung durch: Fahry, P., COBOL programmiert lernen, München 1970. Übungen am Computer, Behandlung von Rechenbeispielen und Beispielen aus der Praxis.
Einführung in die Software 1: Programmiersprachen	Vorlesung eines Praktikers, Unterstützung durch Hersteller-Unterlagen und Übungsaufgaben.
Einführung in die Software 2: Betriebssysteme	Grundlagenliteratur und Vorlesung eines Praktikers, Unterstützung durch Hersteller-Unterlagen.
Einführung in die Software 3: Datenbanksoftware und höhere Prozessoren	Grundlagenliteratur und Vorlesung eines Praktikers, Unterstützung durch Hersteller-Unterlagen.
DV-Anwendungen in der Industrie	Einführung durch Textbuch: Mertens, P., Industrielle Datenverarbeitung, Band 1, 2. Aufl., Wiesbaden 1972. Vorführung von Beispielen aus der Praxis.
DV-Anwendungen in Kreditinstituten und Versicherungen	Grundlagenliteratur und Vorlesung.
DV-Anwendungen in Handel, Verkehr und öffentlichen Diensten	Vorlesung und Gastvorträge.
Computergestützte Informations- und Planungssysteme	Einführung durch Textbuch: Mertens, P. und Griese, J., Industrielle Datenverarbeitung, Band 2, Wiesbaden 1972. Kompaktseminar an einem Wochenende.
Simulation	Einführung durch Textbuch: Mertens, P., Simulation, Stuttgart 1969. Vorlesung.

Lehrveranstaltung	Bemerkungen zur Form
Prognoserechnung	Einführung durch: Schröder, M., Einführung in die kurzfristige Zeitreihenprognose und Vergleich der einzelnen Verfahren, in: Mertens, P. (Hrsg.), Prognoserechnung, Würzburg-Wien 1973. Kompaktseminar an einem Wochenende.
Systemplanung und EDV-Organisation	Grundlagenliteratur, Vorlesung, Diskussion von Fällen, Exkursionen, Gastvorträge.
Seminar zur Betriebsinformatik	Referate der Studierenden, Gastvorträge, Exkursionen. Voraussetzung: Aufnahmeklausur.
Doktoranden-Seminar zur Betriebsinformatik	Doktoranden stellen ihre Arbeitsergebnisse zur Diskussion.

Tab. 4: Ausbildungsplan für das Fach Betriebsinformatik¹⁾

1) Vgl. Mertens, P., Manual ..., a.a.O., S. 2 ff.

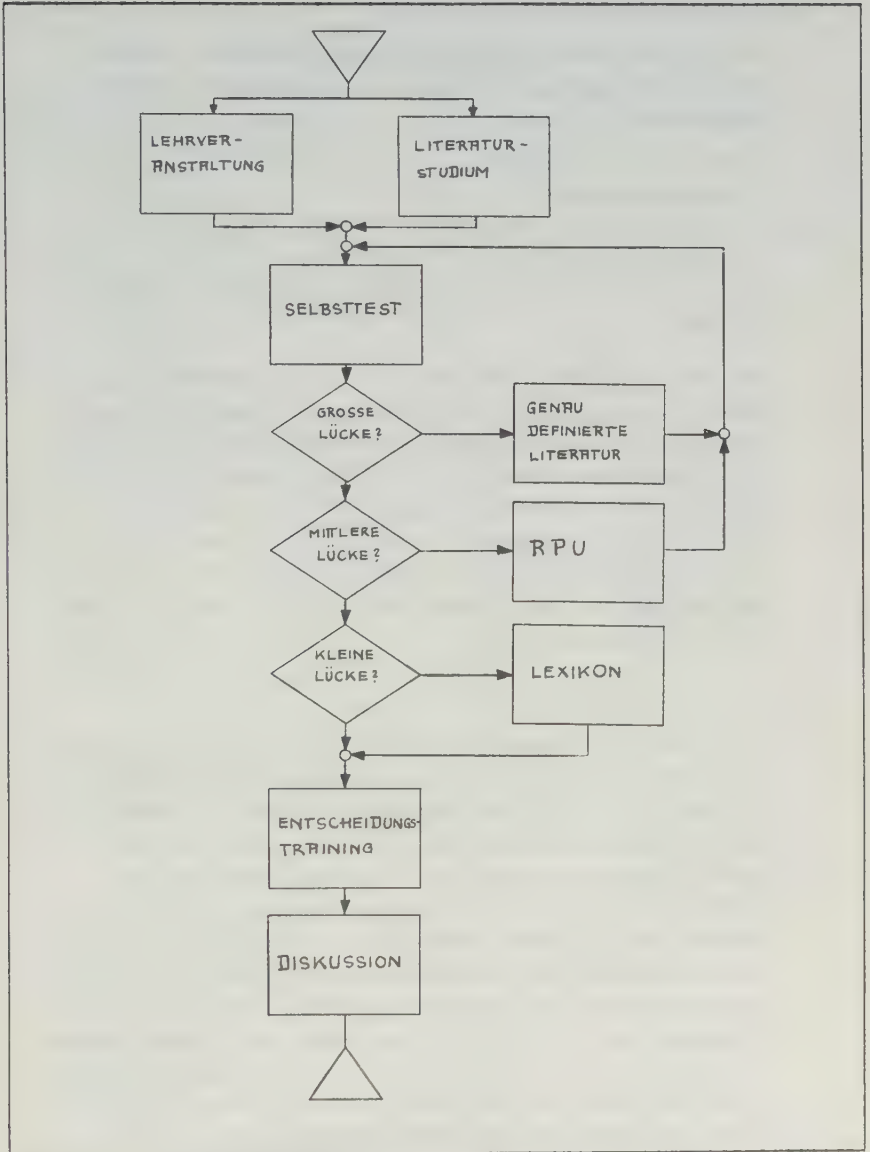


Abb. 8: Ablauf der Lehrveranstaltung "Simulation"

(1) Literaturstudium und Vorlesung

Dem Student steht ein Prospekt zur Verfügung, der für alle im CET enthaltenen Modelle Informationen darüber beinhaltet,

- was ein Modell leistet,
- wie es arbeitet,
- welche Grundkenntnisse für einen sinnvollen Einsatz notwendig sind und
- aus welchen Literaturstellen er sich diese aneignen kann.

Bei der Break-even-Analyse werden beispielsweise Aufsätze und Abschnitte aus Büchern zum Thema der Break-even-Analyse, der Wahrscheinlichkeitstheorie, der Risikoanalyse und der Simulation angegeben. Diese Literaturstellen sollte der Student zur Vorbereitung auf das Training studieren. In der Vorlesung erhält der Lernende Hinweise auf das Arbeiten im Dialog. Er kann zudem eventuell bestehende Unklarheiten über die Modelle beseitigen.

(2) Selbsttests und Training

In einem Selbsttest, der in PUMA programmiert ist, prüft der Student, ob er aufgrund seines Vorwissens bereits sinnvoll mit einem bestimmten Modell arbeiten kann. Bei der Break-even-Analyse erhält er, wenn er nach dem Grundlagenstudium noch Wissenslücken aufweist, vom Rechner abgestufte Literaturhinweise (vgl. Abb. 9); fehlen dem Studenten Grundkenntnisse der Deckungsbeitragsrechnung, der Break-even-Analyse und der Risikoanalyse, so wird er darauf aufmerksam gemacht, daß diese Voraussetzung für das Training sind. Hat er aber Lücken auf dem Gebiet der Simulation, so bekommt er einen Hinweis, daß dieses Wissen zwar wünschenswert, aber nicht notwendig ist. Für besonders interessierte Studenten werden Verweise auf weiterführende Literaturstellen ausgeben¹⁾.

1) Vgl. Rackelmann, G., a.a.O., S. 83 f.

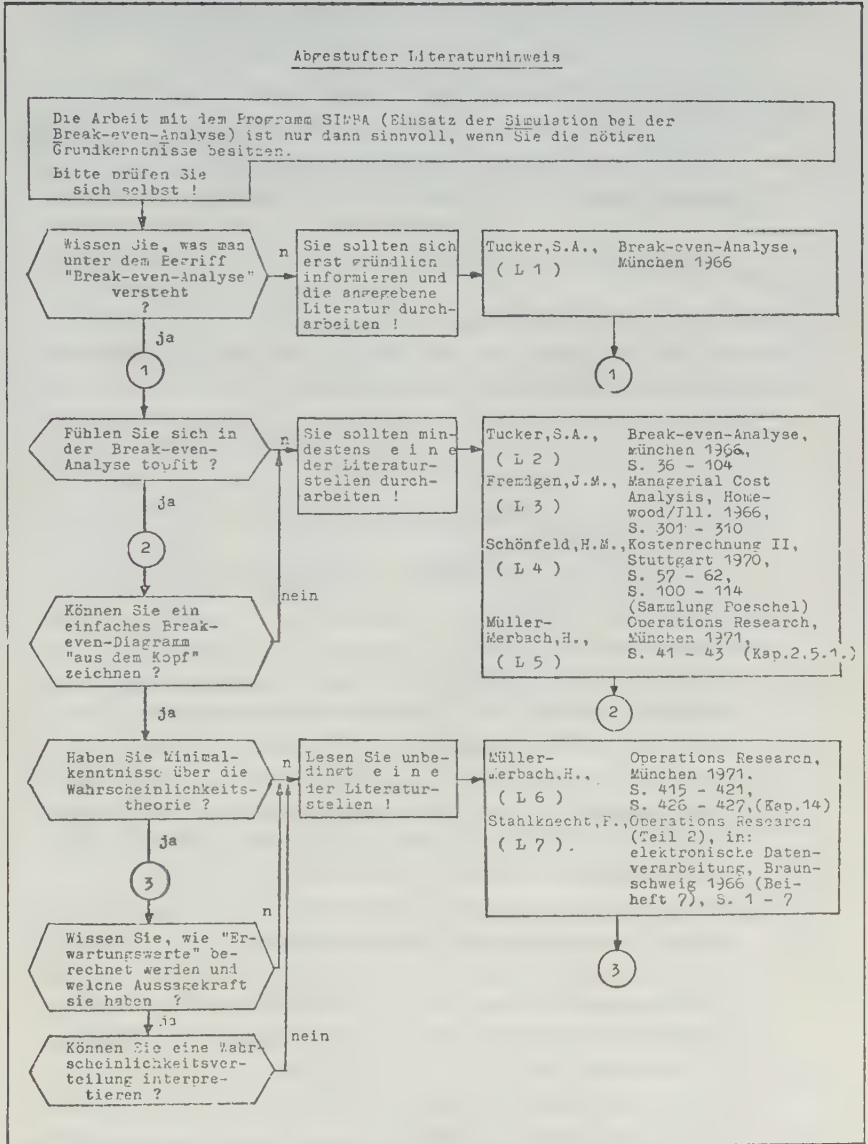


Abb. 9: Ablauf des Selbsttests zum Trainingsprogramm "Break-even-Analyse"¹⁾

1) Entnommen aus: Rackelmann, G., a.a.O., S. 227.

Nach erfolgreichem Test können die Lernenden das Training in Gruppen zu drei Personen absolvieren.

(3) Diskussionen

Anschließend an das Training sind Diskussionen vorgesehen, in denen die Erfahrungen mit dem Lehrstoff ausgetauscht werden können.

1.1.2. Lehrstoff

In der geplanten Lehrveranstaltung arbeitet jeder Student mehrere Trainingsprogramme durch. Er kann dabei aus folgenden Bausteinen auswählen:

(1) Beurteilung von Investitionsvorschlägen mit Hilfe der Simulationstechnik bestehend aus:

- Vorauswahl von Investitionsprojekten aufgrund quantitativer Kriterien,
- Risikoanalyse und Methode der kritischen Werte und
- Nutzwertanalyse.

(2) Werkstättensimulation¹⁾

(3) Risikountersuchung bei der Gewinnschwellenanalyse

(4) Auftragseinplanung bei begrenzten Kapazitäten

(5) Zuteilung von Halbfabrikaten auf Kundenaufträge in der Edelstahlindustrie¹⁾

(6) Steuerpolitik¹⁾

(7) Marktuntersuchung mit Hilfe einer Branchendatenbank¹⁾

(8) Unternehmensgesamtmodell (Schwerpunkt Finanzsphäre)¹⁾

(9) Prognoserechnung mit den Verfahren:

- Exponentielles Glätten erster, zweiter und dritter Ordnung,
- Winters-Algorithmus,
- Schrittweise Regression mit adaptiver Einflußgrößenkombination und
- Vorhersage bei stoßweisem Bedarf¹⁾.

(10) Forschungsplanung mit GERT¹⁾

(11) Netzplanberechnung bei Störungen¹⁾

1) Dieser Baustein ist zum Zeitpunkt der Abgabe dieser Dissertation noch nicht ausgetestet.

1.2. Schwierigkeiten beim Einsatz des CET an der Universität

Die Schwierigkeiten, die einem breiten Einsatz des CET entgegenstehen, lassen sich im wesentlichen in folgenden Punkten zusammenfassen:

- Rechenzeitrestriktionen an den Universitätsrechenzentren,
- hohe Kosten für Dialogstationen,
- häufig ungünstige Lage der Studentendialogplätze und
- der Mangel an CET-Teachware.

Die Begrenzung der Rechenzeit ist derzeit an den meisten Universitäten ein starkes Hindernis für den Einsatz des CET; dieser Engpaß nimmt aber mit der ständig steigenden Leistungsfähigkeit der Rechner immer mehr ab.

Auch die Ausstattung der Universitätsrechenzentren mit den benötigten Dialogstationen ist für das CET meist nicht ausreichend, da die Miete oder der Kauf dieser Geräte sehr teuer ist. Man darf allerdings auch für dieses Problem eine starke Erleichterung erwarten. So kann u.U. die Benutzung von Plasma-Displays (vgl. Teil 3, Punkt 1) bedeutende Kostensenkungen bringen.

Ebenso dürften die Nachteile, die sich aus der örtlichen Bindung der Dialogstationen an ein Rechenzentrum ergeben, durch dezentrale Terminals weitgehend beseitigt werden.

Derzeit gibt es erst für die in Punkt 1.1.2. beschriebenen Gebiete Trainingsprogramme. Der Einsatz des CET ist somit auch von dieser Seite her noch beschränkt. Nur die Entwicklung von CET-Teachware an mehreren Universitäten und ein Austausch der Trainingsprogramme wird es ermöglichen, in verhältnismäßig kurzer Zeit ein breites Angebot von Kursen zu schaffen und den Anteil der Entwicklungskosten an den Kosten pro Terminalsitzung niedrig zu halten.

Obwohl sich die Voraussetzungen für den Einsatz des CET durch die rasche Entwicklung der Computertechnologie verbessern werden, bleibt das CET eine relativ teure Lehrmethode. Man muß daher in absehbarer Zeit weniger trachten, das CET möglichst breit einzusetzen, als vielmehr, das CET als Methode zur Erreichung von Ausbildungszielen zu verwenden, die sonst schwer verwirklicht werden können.

2. Einsatz in der berufsbegleitenden Weiterbildung

Die Opportunitätskosten, die durch das Fehlen am Arbeitsplatz während der Schulungsdauer entstehen, müssen zwar sowohl in der universitären als auch in der beruflichen Weiterbildung beachtet werden, doch ist die Unternehmung wesentlich eher als die Universität bereit, diese Kosten in der Wahl der Ausbildungsmethoden zu berücksichtigen. Es ist für Institutionen der berufsbegleitenden Weiterbildung möglich, teurere Lehrmethoden zu verwenden, wenn dadurch die Ausbildungszeit entsprechend verkürzt werden kann. So setzt beispielsweise I.M.E.D.E. (Institut pour l'Etude des Méthodes de Direction de l'Entreprise) den Rechner bei der Ausbildung von Führungskräften ein, während der RUU auf der Universität derzeit nur experimentell verwendet wird.

Das CET kommt einem Einsatz in der Weiterbildung von Führungskräften in einem wesentlichen Punkt entgegen: Es berücksichtigt in hohem Maße unterschiedliche Adressatenniveaus, indem es die Vorbereitung des Lernenden auf den Trainingsteil steuert. Die Lernziele können somit weitgehend unabhängig vom Ausgangsniveau erreicht werden.

Ein zusätzlicher positiver Effekt ergibt sich aus der Auflockerung des Unterrichts durch die methodische Abwechslung. Dies wird bei Managerseminaren, die meist in Blockform gehalten werden, oft zu einem wesentlichen Argument.

Der Hauptvorteil der Verwendung des CET in diesem Ausbildungsbereich ist die Praxisbezogenheit des Trainings, die den Wissenstransfer auf Problemstellungen der Unternehmung stark begünstigt.

L i t e r a t u r v e r z e i c h n i s
=====

- Alewell, K., Bleicher, K., Hahn, D., Entscheidungsfälle aus der Unternehmenspraxis, Wiesbaden 1971.
- Amedick, W., Simulation des Unternehmerischen Entscheidungsprozesses - dargestellt am Beispiel eines komplexen Unternehmensspieles, Dissertation Berlin o.J.
- Bitzer, D.L., Johnson, R.L., PLATO: A Computer-Based System Used in the Engineering of Education, Control Data Institut, Frankfurt am Main o.J.
- Bode, A., Dütting, M., Computer-Unterstützter Unterricht - Probleme, Autorensprachen, Systemvergleiche, Dokumentation, Diplomarbeit Karlsruhe 1972.
- Brown, R.G., Smoothing, Forecasting and Prediction of Discrete Time Series, Englewood Cliffs 1963.
- Butler, A.K., Frye, C.H., PLANIT Language Reference Manual, SDC-TM-(L)-4422/002/01 Santa Monica 1970.
- Dixon, G.T., Analyzing Student Performance to Improve an Interactive Teaching Program, Massachusetts Institute of Technology, MS-Thesis Cambridge 1970.
- Emmert, P., Investitionsbeurteilung im Wege der Mensch-Maschine-Kommunikation, Dissertation (in Arbeit) Nürnberg.
- Endres, C., PUMA - Programmierter Unterricht unter CD 3300 MASTER, in: Mitteilungsblatt des Rechenzentrums der Universität Nürnberg Nr. 13, Erlangen 1973.
- Faßhauer, R., Mit TOPIC 1 Top-Manager, absatzwirtschaft 14 (1971) 1, S. 17.

- Griese, J. und Matt, G., Prognose mit Hilfe einer Kombination von schrittweiser Regressionsanalyse und exponentieller Glättung (adaptive Einflußgrößenkombination), in: Mertens, P. (Hrsg.), Prognoserechnung, Würzburg-Wien 1973, S. 159 ff.
- Haefner, K., Status und Zukunft des Computerunterstützten Hochschulunterrichts in Naturwissenschaften und Medizin, Arbeitspapier Freiburg 1972.
- Hahn, D., Allgemeine Bedeutung der Fallmethode für die Ausbildung an Hochschulen und in Seminaren für Führungskräfte, in: Alewell, K., Bleicher, K., Hahn, D., Entscheidungsfälle aus der Unternehmenspraxis, Wiesbaden 1971.
- Hedberg, B., On Man-Computer Interaction in Organizational Decision-Making. A Behavioral Approach, Göteborg 1970.
- Holub, G., Probleme beim Einsatz des Computerunterstützten Entscheidungstrainings in der betriebswirtschaftlichen Ausbildung, Dissertation (in Arbeit) Nürnberg.
- IBM (Hrsg.), APL 360-OS and APL 360-DOS, Users Manual, IBM Form GH 20-0906-0.
- IBM (Hrsg.), Coursewriter III, Version 3, Application Description, IBM Form GH 20-0987-2.
- IBM (Hrsg.), IBM Unternehmensplanspiel TOPIC 1, Spielerhandbuch, IBM Form 71485-1.
- Jacob, H. (Hrsg.), Zielprogramm und Entscheidungsprozeß in der Unternehmung, Schriften zur Unternehmensführung, Band 11, Wiesbaden 1970.
- Kosiol, E., Die Behandlung praktischer Fälle im betriebswirtschaftlichen Hochschulunterricht (Case Methode), Berlin 1957.

- Kroll, W., Aufstellen alternativer Investitionsprogramme mit Berücksichtigung des Etats in einem EDV-Dialog-Verfahren, Diplomarbeit Nürnberg 1972.
- Lange, D., Das Unternehmensplanspiel Honeywell Bull OMNILOG III, HB-Schriften 6.
- Langen, H., Weinthaler, F., Prognose mit Hilfe von Verweilzeitverteilungen, in: Mertens, P. (Hrsg.), Prognoserechnung, Würzburg-Wien 1973, S. 113 ff.
- Magee, J.F., Decision Trees for Decision Making, Harvard Business Review 42 (1964) 4, S. 126 ff.
- Mertens, P., Ausbildung in betriebswirtschaftlicher Datenverarbeitung (Betriebsinformatik) an der Wirtschafts- und Sozialwissenschaftlichen Fakultät Erlangen-Nürnberg, Kursmanual, Nürnberg 1973.
- Mertens, P. (Hrsg.), Betriebswirtschaftliche Hochschuldidaktik, Wiesbaden 1972.
- Mertens, P., Der Einfluß der elektronischen Datenverarbeitung auf Entscheidungsfindung und Entscheidungsprozeß, in: Jacob, H. (Hrsg.), Zielprogramm und Entscheidungsprozeß in der Unternehmung, Schriften zur Unternehmensführung, Band 11, Wiesbaden 1970, S. 81 ff.
- Mertens, P., Ergebnisse einer Umfrage zur Verbreitung einzelner Lehr- und Prüfungsformen, in: Mertens, P. (Hrsg.), Betriebswirtschaftliche Hochschuldidaktik, Wiesbaden 1972.
- Mertens, P. (Hrsg.), Prognoserechnung, Würzburg-Wien 1973.
- Mertens, P., Simulation, Stuttgart 1968.
- Mertens, P., Griese, J., Industrielle Datenverarbeitung, Band 2, Informations- und Planungssysteme, Wiesbaden 1972.

- Mertens, P. und Kress, H., Mensch-Maschinen-Kommunikation als Hilfe bei der Entscheidungsvorbereitung und Planung, Zeitschrift für betriebswirtschaftliche Forschung 22 (1970) 1, S. 1 ff.
- Mertens, P. und Plötzeneder, H.D., Ein Experiment mit Programmierem Unterricht im betriebswirtschaftlichen Hochschulstudium, Zeitschrift für Betriebswirtschaft 41 (1971) 3, S. 185 ff.
- Miller, N., The Computers Role in the Educational Process, MIT, MS-Thesis Cambridge 1971.
- Morton, M.S.S., Management-Entscheidungen im Bildschirmdialog, Essen 1972.
- Myers, E., Effective vs. Efficient Computing, Datamation 18 (1972) 6, S. 75 ff.
- o.V., IBGS - Interactive Business Game Simulation, IBM Form GB 21-0393-0.
- OECD CERI (Hrsg.), The Use of Computers in Higher Education, Perspectives and Policies, Technical Report, Centre for Educational Research and Innovation, OECD CERI/CT/70.64.
- Rackelmann, G., Einsatz der Simulation bei der Break-even-Analyse, Diplomarbeit Nürnberg 1973.
- Reitbauer, F., Das Trainingssystem im Computerunterstützten Entscheidungstraining, Dissertation (in Arbeit) Nürnberg.
- Rockart, J.F., Morton, M.S.S., Zannetos, Z.S., Associative Learning Project - Phase 1 System, Massachusetts Institut of Technology, Working Paper 440-70 Cambridge 1970.
- Rockart, J.F., An Integrated Use of Available Resources (Student, Professor and Technology) in the Learning Process, MIT, Working Paper 601-72 Cambridge 1972.

- Rockart, J.F., o.T., Massachusetts Institute of Technology, Arbeitspapier Cambridge 1972.
- Schießl, W., Risikoanalyse von Investitionsvorhaben im Rahmen eines EDV-Dialog-Modells, Diplomarbeit Nürnberg 1972.
- Schmitt, A., Zum Entwurf eines flexiblen und portablen Software-Systems für Rechnergestützten Unterricht, Institut für Informatik der Universität Karlsruhe, Interner Bericht Nr. 7 Karlsruhe 1972.
- Schneider, H., Beurteilung qualitativer Einflußgrößen bei Investitionsobjekten in einem EDV-Dialogmodell mit Berücksichtigung einer durch quantitative Kriterien getroffenen Vorauswahl, Diplomarbeit Nürnberg 1972.
- Siemens (Hrsg.), Siemens System 4004 LIDIA, Autorensprache für rechnergestützte Unterweisung, Beschreibung, D14/4005.
- Steinmann, H., Kumar, B., Bleyer, E., Die Fallmethode in der universitären Führungsausbildung - Ergebnisse einer empirischen Befragung betriebswirtschaftlicher Lehrstühle, Universität Erlangen-Nürnberg, Arbeitspapier Nürnberg 1972.
- Stubenrauch, H., DELPHI - ein handelsorientiertes Unternehmensspiel, data praxis, Bestell-Nr. 2-2600-829.
- Stumpf, R., Erfassung und Zusammenspiel quantitativer Einflußgrößen bei Investitionsvorhaben in einem computerunterstützten Simulationsmodell, Diplomarbeit Nürnberg 1972.
- Thorelli, H.B., Graves, R.L., Howells, L.T., Spielerhandbuch zum INTOP-Planspiel, Indiana University, Bloomington 1966.
- Wahi, P.N., A general management business simulation in APL, IBM Systems Journal 11 (1972) 2, S. 269 f.
- Weltner, K., Informationstheorie und Erziehungswissenschaft, Quickborn 1970.
- Willmann-Institut (Hrsg.), Lexikon der Kybernetik, Bd. 4, Freiburg 1971.

Wolf, F., Seybold, G., EDI: Beschreibung eines einfachen Dialog-Interpreters für die CD 3300, Universität Erlangen-Nürnberg, Rechenzentrum, Interner Arbeitsbericht Nr. 26 Erlangen 1973.

Zinn, K., Requirements for Programming Languages in Computer-Based Instructional Systems; Technical Report; Organization für Economic Co-operation and Development (OECD); Centre for Educational Research and Innovation, CERI/CT/70.62 Paris 1971.

C U R R I C U L U M V I T A E

24. 6. 1949 Geburt in Dornbirn (Österreich)
- 1955 - 1959 Volksschule (Wolfurt)
- 1959 - 1967 Humanistisches Bundesgymnasium (Bregenz)
- WS 1967/68 Studium der Betriebswirtschaftslehre an
der Universität Innsbruck
- SS 1968 - SS 1971 Studium der Betriebswirtschaftslehre an
der Hochschule Linz
13. 6. 1971 Zweite Diplomprüfung abgelegt und damit
den akademischen Grad Magister der Sozial-
und Wirtschaftswissenschaften (Mag. rer.
soc. oec. H.Ö.) erworben
- 1970 Wissenschaftliche Hilfskraft am Institut für
Fertigungswirtschaft und betriebliche System-
forschung
- 1971 - 1973 Doktoratsstudium an der Friedrich-Alexander-
Universität Erlangen-Nürnberg
- 1971 - 1973 Verwalter der Dienstgeschäfte eines wissen-
schaftlichen Assistenten am Lehrstuhl für
Betriebswirtschaftslehre, insbesondere für
Betriebs- und Wirtschaftsinformatik an der
Friedrich-Alexander-Universität Erlangen-
Nürnberg
7. 11. 1973 Promotion

Offsetdruck-Fotodruck

3. Hög, 8520 Erlangen, Hauptstr. 109 (Nähe Martin-Luther-Platz) Tel. 22990